



TITLE:

量子スピン系の基底状態を有限系 ($N \leq 20$)について計算するプログラム ム

AUTHOR(S):

田口, 善弘; 西森, 秀稔

CITATION:

田口, 善弘 ...[et al]. 量子スピン系の基底状態を有限系($N \leq 20$)について
計算するプログラム. 物性研究 1986, 45(5): 299-349

ISSUE DATE:

1986-02-20

URL:

<http://hdl.handle.net/2433/91889>

RIGHT:

量子スピン系の基底状態を有限系 ($N \leq 20$) について計算するプログラム

東工大 理 田口善弘 西森秀稔

(1985年12月26日受理)

§ 0. はじめに

我々は、 $S=1/2$ 量子スピン系の基底状態に興味を持ち、一連の研究を行った。¹⁾ その際、使用した「有限系の基底状態の数値的厳密解」を求めるプログラムは、非常に汎用性が高く、 $S=1/2$ の量子スピン系でありさえすれば、格子、次元、異方性のいかにかわらず、即座に適用できるような仕様となっている。このプログラムを多くの人々につかってもらうべく解説する。内容はともかく、すぐ使いたいという方は、Appendix. A~Fを参照されたい。(なお、このプログラム・パッケージは、スピン総数 $N \leq 20$ までとなっているが、 $N=21$ ができるように直すことも可能である。詳細については、問い合わせられたい。)

§ 1. モデル解説

我々のプログラムで解くことができるのは、次のようなハミルトニアンで記述できる $S=1/2$ の量子スピン系である。

$$\mathcal{H} = -2 \sum_{\langle i, j \rangle} J_{ij} \{ \Delta_{ij} S_i^z S_j^z + 1/2 (S_i^+ S_j^- + S_i^- S_j^+) \} \quad (1)$$

但し、相互作用している site のペア、 $\langle i, j \rangle$ はいかなるものであっても構わない (nearest neighbor, next nearest neighbor, infinite range, etc.)。求めることができるのは、基底状態、及び、低次励起状態の固有エネルギー、及び、固有状態である。縮退していてもよい。

§ 2. プログラム化への指針 1 (行列の生成)

我々は、ハミルトニアン (1) を解く為に、これを行列の固有値問題として扱った。即ち、行列の対角化を数値的に行なうわけである。まず、適当な正規完全直交関数系に対する、行列要素を求めなくてはならない。我々が、採用した正規完全直交関数系は、 N 個のスピン成分を対角化する表示の直積よりなる波動関数、つまり、 $|\uparrow\downarrow\downarrow\uparrow\cdots\uparrow\downarrow\rangle$ である。このような波動関数は 2^N 個ある。これを 2 進数として表現し、 $0 \sim 2^N - 1$, で番号付けをすることにした。例えば、 N を系の大きさ (スピンの総数) とすると、 $N=4$; $|\uparrow\uparrow\downarrow\uparrow\rangle = |1101\rangle$ (2 進法) $= |13\rangle$ (10 進法) となる。これにより行列要素を簡単に求め、順序よく並べることができる。 $N=4$ の 1 次元強磁性ハイゼンベルグ・スピンを例にとると (n. n. 相互作用、周期的境界条件) ハミルトニアンは、

$$\mathcal{H} = -2J \sum_{i=1}^4 \mathbf{S}_i \cdot \mathbf{S}_{i+1} \quad (2)$$

と表現できる。(ここで、2 進表示での site の番号付けは下の位から、1, 2, 3, 4, とすることにする。) この場合、例えば行列要素 $\langle 1 | \mathcal{H} | 8 \rangle$ は、

$$\langle 1 | \mathcal{H} | 8 \rangle = \langle 0001 | \mathcal{H} | 1000 \rangle = \langle \downarrow\downarrow\downarrow\uparrow | \mathcal{H} | \uparrow\downarrow\downarrow\downarrow \rangle = -J$$

となる。この様にしてこの正規完全直交関数系に対するハミルトニアン行列 $\langle n | \mathcal{H} | m \rangle$, $n, m = 0, 1, \dots, 2^N - 1$ を一般の N に対して作ることができた。後はこれを対角化すればよい。まともに対角化すると、 $2^N \times 2^N$ の行列を対角化しなくてはならないので、我々は、自明な対称性を 1 つだけ用いて行列の次元を下げた。すぐに解るように (1) 式のハミルトニアン \mathcal{H} は、 z 方向の全スピン量子数演算子 S_{tot}^z と可換である。言い換えれば、 S_{tot}^z ごとにハミルトニアン行列 $\langle n | \mathcal{H} | m \rangle$ がブロック対角化されているのである。それ故、我々は S_{tot}^z の値ごとに分類されたハミルトニアン行列を対角化することにした。具体的には、 $N=4$ の時は、 16×16 の行列を対角化するかわりに、 $S_{\text{tot}}^z = 0$; 6×6 , $S_{\text{tot}}^z = 1$; 4×4 , $S_{\text{tot}}^z = 2$; 1×1 , の 3 種の行列を独立に対角化するのである。 S_{tot}^z によるこのような分類は、なんら汎用性を損なうものではなく、むしろ、 S_{tot}^z の固有関数であるという事実は磁化などを考える際には大変便利である。

§ 3. 行列の対角化

— LANCZOS法とCG法 —

我々のアルゴリズムを説明するためには、18万次元 ($N=20$ の時の $S_{\text{tot}}^z=0$ の部分空間の次元数) という大規模行列の対角化を可能にした2つの手法について解説する必要がある。それはLANCZOS法とCG法である。

a) 固有値

まず、固有値を求める場合であるが、それは以下の様な手順による。

LANCZOS法

BI-SECTION 法

(ハミルトニアン行列) \Rightarrow (三重対角化) \Rightarrow (対角化)

我々が開発したのは、ランチョス法の部分であり、バイセクション法はHITACのできあいのライブラリーを使用した。よってここではランチョス法のごく簡単な解説をしておく。ランチョス法は反復法的一种であり、与えられた適当な初期ベクトルに対して反復を繰り返し、三重対角行列を得る。現実には誤差が積み重なるので、数百次元程度でも全ての固有値を正確に与えるような三重対角行列を求めることは難しい。しかし、その反面、最低固有値のみ、あるいは、その近傍の固有値を求めるだけであれば、三重対角化を部分的にするだけで、正確な固有値を与えるような三重対角行列を求めることができる。(経験的にはハミルトニアン行列の次元数によらず(18万次元のものであっても) わずか、50~60次元程度の三重対角化で、大変よい精度の値を得ることができるようである。) 次に述べる具体的な手順を見ていただければ理解しやすいであろう。

ランチョス法は反復法なので、普通の対角化の手法と異なり、行列を変形したりはしない。そのかわりに、与えられた初期ベクトルに対して反復が繰り返されるごとに、三重対角行列の対角要素 α_i と、準対角要素 β_i が、順々に求まって行くのである。つまり、

$$[A_{ij}] \xrightarrow{\text{反復1}} \begin{bmatrix} \alpha_1 & \beta_1 & & 0 \\ \beta_1 & & & \\ 0 & & & \\ & & & \end{bmatrix} \xrightarrow{\text{反復2}} \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \beta_2 & 0 \\ & \beta_2 & & \\ 0 & & & \end{bmatrix} \longrightarrow \dots$$

その際、第 i 回目の反復ベクトルを v_i とすると、 α_i , β_i , v_i の間には、次の様な簡単な関係があることが知られている。(参考文献(ref.2) p169 アルゴリズム①~⑤)

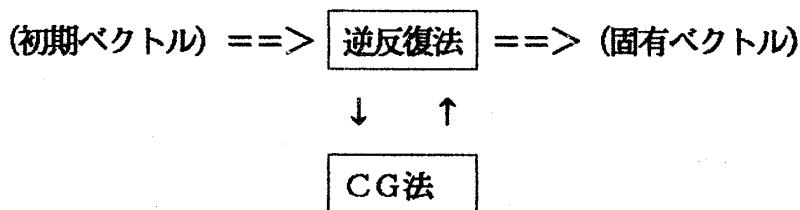
$$\alpha_i = v_i^T \cdot \lambda \cdot v_i, \quad \beta_i = \| \lambda \cdot v_i - \beta_{i-1} \cdot v_{i-1} - \alpha_i \cdot v_i \|$$

$$v_i = (v_{i-1} - \beta_{i-2} \cdot v_{i-2} - \alpha_{i-1} \cdot v_{i-1}) / \beta_{i-1}$$

もとの行列 λ を変形しなくてよいというところがミソである。しかし、すぐに解るように α_i , β_i の値は、出発ベクトル v_1 をどうとるかによってかわってくる。正しい固有値を与えるような α_i , β_i の値を得るためには、 v_1 は求めるべき固有値に対応する固有ベクトルに対して有限の射影を持っていなければならないが、一般には事前にそんなことは解らないので、 v_1 を数通りとってみる必要がある。我々の経験では2通りで充分だと思われるが、心配ならば最初のうちは5通りくらいとってみてもよいであろう。とにかく、このようにして α_i , β_i を順番に求めるわけである。具体的には、 α_i , β_i の数を逐次増やしながら、その都度、それらよりなる三重対角行列の対角化を実行し、必要な固有値の収束が確認されたらやめる、というふうに行っている。但し、LANCZOS法では、固有値の縮退の有無は判定できない。それを知るためには、固有ベクトルまで求めてみるしかない。

b) 固有ベクトル

固有ベクトルを求める場合の手順は次の通りである。



まず、逆反復法について解説しよう。初期ベクトル ϕ は、行列 λ の固有ベクトル ϕ_i で必ず展開できる。 $\phi = \sum C_i \cdot \phi_i$, $\lambda \cdot \phi_i = \epsilon_i \cdot \phi_i$

n 番目の固有値 ϵ_n に対応する固有ベクトル ϕ_n を求めたいとしよう。

$(\lambda - \epsilon_n \cdot I)^{-1} \cdot \phi = \sum C_i / (\epsilon_i - \epsilon_n) \cdot \phi_i$ なので、 $(\lambda - \epsilon_n \cdot I)^{-1}$ を ϕ に作用させると、 ϕ に含まれている固有ベクトルのうち、 ϕ_n だけが、無限大の相対倍率で取り出されることが解る。 ϵ_n が、厳密な固有値であれば、結果は発散してしまうが、実際は数値解を入力するので必ず誤差 $\Delta \epsilon$ がふくまれている。つまり、実際の増幅率は $1/\Delta \epsilon$ 程度ということになる。しかし $(\lambda - \epsilon_n \cdot I)^{-1}$ を繰り返し作用すれば、初期ベクトル ϕ から、 ϕ_n 以外の固有ベクトルがほどなく消え去ってしまうであろうことは想像に難くない。実際、 $N=20$, 18万次元の場合ですら、わずか数回 $(\lambda - \epsilon_n \cdot I)^{-1}$ を作用させるだけで、 ϕ から ϕ_n が抽出されている。こう書くと簡単なようだが、よく考えると $(\lambda - \epsilon_n \cdot I)^{-1}$ は求まらないのである。逆行列を求めるだけのメモリーがとれるなら最初から、LANCZOS法など用いずに固有値が求められる。それができないから苦労しているわけであって、まるで解決になっていない。そこで、一工夫してやる。つまり、 $(\lambda - \epsilon_n \cdot I)^{-1} \cdot \phi$ が求まればよいのであるから、 $\phi^{(1)} = (\lambda - \epsilon_n \cdot I)^{-1} \cdot \phi$ とおいてやって、 $(\lambda - \epsilon_n \cdot I) \cdot$

$\phi^{(1)} = \phi$ と書き直し、これを $\phi^{(1)}$ について解いてやるのである。CG法を用いればこのような一次方程式系を、逆行列を求めることなく解くことができる。そして、更に、 $(\lambda - \varepsilon_n \cdot I) \cdot \phi^{(2)} = \phi^{(1)}$, $(\lambda - \varepsilon_n \cdot I) \cdot \phi^{(3)} = \phi^{(2)}$, \dots という風に、次々と $\phi^{(i)}$ を求めてやれば、これは $(\lambda - \varepsilon_n \cdot I)^{-1}$ を何回も作用させたことに等しい。これを、収束するまでやればいいのである。以下に、CG法の考え方を簡単に述べておく。CG法は、本質的には、極値問題の解法である。一次方程式 $A \cdot x = b$ を解くという問題は、残差 $r = A \cdot x - b$ のノルム $\|r\|$ を最小にする (つまり、ゼロ) ような x を求める問題と見ることが出来る。そこで、残差のノルム $\|r\|$ を x の関数と見て、 $f(x) = \|r\|$ とおき、 $f(x)$ をポテンシャルとみなして、適当な初期ベクトル x_0 から出発して、 $f(x)$ 最小の点に到達するまでポテンシャル面上を移動して行くのが、CG法のもともとの発想である。 $f(x)$ は、逆行列を求めなくても書き下せるから、逆行列は求めないですむというわけである。詳細は参考文献 (ref. 2) を見て欲しい。我々のプログラムの方で記号の対応まで合わせてあるのですぐ解るはずである。具体的には、参考文献 p 95 ~ p 97, p 95 頁末から、p 96 にかけて書かれているアルゴリズム①~⑦に従っている。但し、④のかわりに、p 97 の④' を用いており、プログラムでは $A \cdot P_k = Y_k$ と表記している。参考文献にもある通り、CG法の収束速度はきわめてよく、もともとの A の次元数が何万次元という場合でも、数十回の反復で収束している。我々の経験では例外はただ1つあったのみ (fcc 格子上の最近接相互作用、反強磁性ハイゼンベルグ模型における、 $N=20$ の場合) で、殆どの場合、充分、実用にたえるだけの収束速度が得られた。最後に、一つだけ注意を述べておく。逆反復法は、求めるべき固有状態が ε_n に対して縮退している時は、初期ベクトルの値によって、縮退している固有ベクトルのある線型結合を与えることになる。この場合は、初期ベクトルをいくつか変えて、複数の固有ベクトルを求め、これを直交化することによって、いくつ独立なベクトルがあるかを調べなくてはならない。(我々のプログラム・パッケージには、この為のサブルーチン、ORTHGも用意されている) 勿論、縮退しているかどうかは事前には解らないので、必ず複数の初期ベクトルで試さなければならない。この複数のベクトルで試すという操作は、反復法のつねとしてある求めるべき固有値に対応しない固有ベクトルに収束してしまうという危険をさけるためにも、省くことのできない手続きである (求めるべき固有状態が得られたかどうかは、サブルーチン CHECK を用いて確認すればよい)。なお、この方法によって求められた固有ベクトルの各係数は7~8桁の精度を持っていることが、経験的に、確認されている。

§ 4. プログラム化への指針 2 (行列とベクトルの積の計算)

前置きが長くなってしまったが、ようやく、我々の作ったプログラムの一番、本質的な部分について述べたいと思う。その為には、なぜ、LANCZOS法やCG法の活用が有用であるのかという、今まであからさまには述べてこなかった点について述べなくてはならない。それは次の一言に要約される。「もとのハミルトニアン行列が大規模疎行列（つまり、要素の大部分がゼロ）であること」。普通の対角化の手法、消去法などでは、行列の基本変形を行なうため、行列の要素の総数に等しい作業領域が必要である。ところが、§ 3を見れば解るように、LANCZOS法やCG法では、基本変形をする必要がなく、ただ単に与えられたハミルトニアン行列 λ と任意のベクトル ϕ の積 $\lambda \cdot \phi$ が計算できればよい。ハミルトニアン行列の大部分がゼロであるこの問題のばあいには、ノンゼロの要素の位置と値のみ記憶しておけばよいことになり、要素の総数に比べて著しく小さい作業領域のみで済むことになる。我々が、LANCZOS法やCG法に目をつけたのは、この故である。しかし、ここで更に一工夫すれば必要とする作業領域をより少なくすることができる。それは、原子核理論の分野では既に有名な手法、「ハミルトニアン行列を作りながら、 $\lambda \cdot \phi$ を計算する」という手法である。この手法を使えば、 λ を記憶する領域さえ不要となり、ベクトル ϕ のための作業領域のみで済ますことができる。我々のアルゴリズムは、これに従っている。(2)式のハミルトニアンを $S_{\text{tot}}^z = 0$ の空間で対角化する場合を例にとって、具体的にプログラムに即して (Appendix. D参照) 解説することにしよう。我々はまず、 $S_{\text{tot}}^z = 0$ に属するベクトルの種類を知らなければならない。それは、 $N=4$ の場合、§ 2の表現で言うと、 $|3\rangle$, $|5\rangle$, $|6\rangle$, $|9\rangle$, $|10\rangle$, $|12\rangle$, の6つであり、これが、LIST1という配列に蓄えられている。更に、これらが、逆に、何番目のベクトルであるかも解らなくてははいけない。つまり、 $|3\rangle$ が、LIST1の何番目であるかを覚えておかななくてははいけない。これが、LIST2という配列に蓄えられている。すなわち、 $\text{LIST2}(3) = 1$, $\text{LIST2}(5) = 2$, \dots , $\text{LIST2}(12) = 6$, である。このLIST1とLIST2という配列があると、行列を作りながら行列とベクトルの積 $\lambda \cdot \phi$ を計算することができる。

$$\begin{aligned}\phi &= C_a |3\rangle + C_b |5\rangle + C_c |6\rangle + C_d |9\rangle + C_e |10\rangle + C_f |12\rangle \\ &= \sum_i' C_i |n_i\rangle\end{aligned}$$

(\sum' は、 $i = a \sim d$ の和)

$$\text{であるとする、}\lambda \cdot \phi = \sum_j \{ \sum_i' \langle n_j | \lambda | n_i \rangle \cdot C_i \} |n_j\rangle$$

と計算できる。これを、プログラムでは次の様に行っている。まず、 ϕ を表現する一次元配列、 $v = (Ca, Cb, Cc, Cd, Ce, Cf)$ を作る。次に、 λ のうち、とある1つの項、たとえば、 $S_2 \cdot S_3$ を選択する。そして、配列 v の第一要素、の値を見る。これは、 Ca である。ついで、LIST1 (1)を見ると、3が入っている。それで、 Ca の重みを持つのは $|3\rangle$ という状態であると解る。 $|3\rangle$ とはいかなる状態であるかということ、 $|3\rangle = |0011\rangle = |\downarrow\downarrow\uparrow\uparrow\rangle$ である。つまり、site 1と2がup, 3と4がdownという状態である。これに、 $S_2 \cdot S_3 = S_2^z \cdot S_3^z + 1/2 (S_2^+ \cdot S_3^- + S_2^- \cdot S_3^+)$ を作用させると、 $S_2 \cdot S_3 |3\rangle = -1/4 \cdot |3\rangle + 1/2 \cdot |5\rangle$ となる。そこで行列要素を持つのはブラが $\langle 3|$ と $\langle 5|$ の時のみであると解る。 $\langle 3| S_2 \cdot S_3 |3\rangle = -1/4$, $\langle 5| S_2 \cdot S_3 |3\rangle = 1/2$ 。そこで、これを $\lambda \cdot \phi$ の為に確保された配列 v' に代入する。 $\langle 3| S_2 \cdot S_3 |3\rangle$ は対角要素であるから、問題はない。 $-1/4 Ca$ を v' の第一要素に加える。問題は、 $\langle 5| S_2 \cdot S_3 |3\rangle$ の方である。 $1/2 Ca$ は、 $\lambda \cdot \phi$ の $|5\rangle$ に相当する v' の要素に加えなくてはいけない。しかし、 $|5\rangle$ が第何番目の要素であるかは解らない。そこでLIST2 (5)を見る。するとLIST2 (5) = 2であるので、 $|5\rangle$ は2番目の要素であることが解る。そこで v' の第二要素に $1/2 Ca$ を加える。後はこのくりかえしである。 v の第2~6要素について全く同じことを行い、それが終わったら、今度は、 λ のうちの別の項、 $S_1 \cdot S_4$ etc. についても全く同じことを行なう。これらの行程を全部終了すると $\lambda \cdot \phi$ を表現する配列 v' が求まるというわけである。我々のプログラムでは更に高速化を図るため、 $S_1 \cdot S_2 |3\rangle$ という演算を、2進法のままbit演算で行っている。その詳細はAppendix. H を見られたい。

§ 5. おわりに

以上で、我々のプログラムの解説は終わりである。AppendixにLANCZOS法、逆反復法、CG法のフロー・チャート、使用の手引き、bit演算の解説、および、プログラムリスト、エラー・メッセージの説明をつけておいた。なお、プログラムへのコメント等ありましたら、是非、お知らせください。最後に、このプログラムを用いた研究には小口武彦先生が色々ご指導下さったことを感謝を込めて、述べておきます。

【参考文献】

- ① T. Oguchi, H. Nishimori, Y. Taguchi, J Phys. Soc. Jpn. 55 (1986) NO. 1, NO. 2, 54 (1985) NO. 12
- ② 戸川隼人 「マトリクスの数値計算」 オーム社

Appendix. A

プログラム・パッケージ "TITPACK" マニュアル

(Appendix. Bをあわせて参照のこと)

本プログラム・パッケージはHITAC S-810/20上で
ライブラリー・プログラム MATRIX/HAPを用いて
走らせなくてはなりません。

ご注意ください。

本プログラム・パッケージで対角化できるのは

$$\lambda = -2 \sum_{\langle i, j \rangle} J_{ij} \{ \Delta_{ij} S_i^z S_j^z + 1/2 (S_i^+ S_j^- + S_i^- S_j^+) \}$$

なる形で書ける、 $S=1/2$ 、 $N \leq 20$ のスピンのハミルトニアンである。実行にあたっては、次の様なパラメーターを与えてやらねばならない。

N:スピンの総数

MAX:行列の次元 (N, SZVALによる)

IBOND: $\langle i, j \rangle$ の総数

IPAIR: $\langle i, j \rangle$ の内容

BONDWT: J_{ij}

ZRATIO: Δ_{ij}

ISTEP:収束確認頻度

IV:初期ベクトルのノンゼロ要素の位置

SZVAL:対角化したい空間の S_{tot}^z の値

(注)

@...パラメーターとして与えるべきもの

#...計算結果

(1) LANCZS ; 固有値を求める

CALL LANCZS(N, IPAIR, IBOND, V2, V1, V0, DELM, ZRTIO, ISTEP, E, I, IV, LIST1, LIST2, MAX,
BONDWT)

@ N	整数型	スピンの総数 ($N \leq 20$)
@ IPAIR(2*IBOND)	整数型	相互作用している site のペア (ボンド) ($2n-1$ 番目と $2n$ 番目 ($n=1, 2, 3, \dots$) の要素が相互作用しているように 代入すること)
@ IBOND	整数型	ボンドの総数
V2 (MAX)		
V1 (MAX)	倍精度実数型	作業領域 ($MAX \leq 184756$)
V0 (MAX)		
DELM (MAX)		
@ ZRTIO(IBOND)	倍精度実数型	異方性 Δ_{ij} (IPAIR の要素と同じ順序になるように、 代入すること。つまり、ZRTIO(n)は、 IPAIR(2n-1)-IPAIR(2n) の相互作用)
@ ISTEP	整数型	収束を、確認する頻度 (ISTEPに1回) (我々の経験では ISTEP \div 5 でよい)
# E (4)	倍精度実数型	何も入れなくてよい。求められた固有値の 値が代入されて、返される。 (E(1)が基底状態。E(2)~E(4)は、第2~第4 励起状態。E(2)までは収束している。 エネルギーの単位は J_{ij} とおなじ。)

# I	整数型	何もいれなくてよい。 (収束するまでに要した、ITERATION の回数 が入って戻ってくる。)
@ IV	整数型	初期ベクトルの何番目の要素を1にするか。 ($1 \leq IV \leq MAX$ のどれでもよい) (LANCZOS法は反復法なので、IVの値に よっては、E(1)が基底エネルギーに収束しな いこともありうるので、IVの値を、2~3通 りかえて見て、そのうち、1番低いE(1)の値 をとる必要がある。)
@ LIST1(MAX)	整数型	n番目の要素がいかなる状態であるかを、 入れておく。(→サブルーチンSZで求める。)
@ LIST2(2*N)	整数型	いかなる状態が、何番目の要素であるかを 入れておく。(→サブルーチンSZで求める。)
@ MAX	整数型	行列の次元数 $MAX = \begin{matrix} N \\ C \\ M \end{matrix}, M = N/2 - S_{tot}^z$ $(S_{tot}^z = m/2, m=0, 2, 4, \dots \quad N: \text{even}$ $m=1, 3, 5, \dots \quad N: \text{odd})$
@ BONDWT(IBOND)	倍精度実数型	相互作用 J_{ij} (代入の仕方はZRTIO に同じ。)

(2) INVITR ; 逆反復法で固有ベクトルを求める。

CALL INVITR(E, IPAIR, IBOND, LIST1, LIST2, MAX, N, B, X, P, R, Y, ISTEP, IV, ZRTIO, BONDWT)

@ E	倍精度実数型	求める固有ベクトルに対応する固有値 (LANCZSで求めたE(1), E(2) etc)
-----	--------	---

@ IPAIR (2*IBOND)	整数型	相互作用している site のペア (ボンド) ($2n-1$ 番目と $2n$ 番目 ($n=1, 2, 3, \dots$) の要素が相互作用しているように代入すること)
@ IBOND	整数型	ボンドの総数
@ LIST1 (MAX)	整数型	n 番目の要素がいかなる状態であるかを、 入れておく。(→サブルーチン SZ で求める。)
@ LIST2 (2*N)	整数型	いかなる状態が、何番目の要素であるかを 入れておく。(→サブルーチン SZ で求める。)
@ MAX	整数型	行列の次元数 $MAX = \sum_M C_M, \quad M = N/2 - S_{tot}^z$ $(S_{tot}^z = m/2, m=0, 2, 4, \dots \quad N: \text{even}$ $m=1, 3, 5, \dots \quad N: \text{odd})$
@ N	整数型	スピンの総数 ($N \leq 20$)
B (MAX)	倍精度実数型	作業領域 ($MAX \leq 184756$)
# X (MAX)	倍精度実数型	何も入れなくてよい。 (求まった固有ベクトルが入って戻って 来る。)
P (MAX)		
R (MAX)	倍精度実数型	作業領域
Y (MAX)		
@ ISTEP	整数型	収束を、確認する頻度 (ISTEP に 1 回) (我々の経験では ISTEP \div 5 でよい)

@ IV	整数型	初期ベクトルの何番目の要素を1にするか。 ($1 \leq IV \leq \text{MAX}$ のどれでもよい) (逆反復法は反復法的一种なので、IVの値によっては、固有ベクトルが、固有値Eに対応する固有ベクトルに収束しないこともありうるので、サブルーチン CHECKを用いて正しい固有値をもつ固有ベクトルに収束したかどうかを確認する必要がある。さらに、縮退度まで知る為には、IVの値を、2~3通りかえて見て、得られた固有ベクトルをORTH Gで、直交化してみなくてはならない)
@ ZRTIO(IBOND)	倍精度実数型	異方性 $\Delta i j$ (IPAIR の要素と同じ順序になるように、代入すること。つまり、ZRTIO(n)は、IPAIR(2n-1)-IPAIR(2n) の相互作用)
@ BONDWT (IBOND)	倍精度実数型	相互作用 $J i j$ (代入の仕方はZRTIO に同じ。)

(3) CG ; 逆反復法の逆行列演算を行なう。

CALL CG(E, IPAIR, IBOND, LIST1, LIST2, MAX, N, B, X, P, R, Y, ISTEP, ITERAT, ZRTIO, BONDWT)

@ E	倍精度実数型	求める固有ベクトルに対応する固有値 (LANCZSで求めたE(1), E(2) etc)
@ IPAIR(2*IBOND)	整数型	相互作用している site のペア (bond) (2n-1番目と2n番目 (n=1, 2, 3, ...) の要素が相互作用しているように代入すること)

@ IBOND	整数型	ボンドの総数
@ LIST1 (MAX)	整数型	n 番目の要素がいかなる状態であるかを、 入れておく。(→サブルーチン S Z で求める。)
@ LIST2 (2**N)	整数型	いかなる状態が、何番目の要素であるかを 入れておく。(→サブルーチン S Z で求める。)
@ MAX	整数型	行列の次元数 $\text{MAX} = {}_N C_M, M = N/2 - S_{\text{tot}}^z$ $(S_{\text{tot}}^z = m/2, m = 0, 2, 4, \dots \quad N: \text{even}$ $1, 3, 5, \dots \quad N: \text{odd})$
@ N	整数型	スピンの総数 ($N \leq 20$)
B (MAX)	倍精度実数型	連立方程式の右辺 (§ 3 でいうと、1 つ 前の近似ベクトル $\phi^{(i-1)}$)
# X (MAX)	倍精度実数型	何も入れなくてよい。 (新たな近似ベクトル $\phi^{(i)}$ が入って 戻ってくる。)
P (MAX)	倍精度実数型	作業領域
R (MAX)		
Y (MAX)		
@ ISTEP	整数型	収束を、確認する頻度 (ISTEP に 1 回) (我々の経験では、ISTEP \div 5 でよい)
# ITERAT	整数型	収束に要した反復回数
@ ZRTIO (IBOND)	倍精度実数型	異方性 Δ_{ij} (IPAIR の要素と同じ順序になるように、

代入すること。つまり、 $ZRTIO(n)$ は、 $IPAIR(2n-1)-IPAIR(2n)$ の相互作用)

@ BONDWT (IBOND) 倍精度実数型 相互作用 J_{ij}
(代入の仕方はZRTIO に同じ。)

(4) SZ ; LIST1, LIST2を作る。

CALL SZ (LIST1, LIST2, MAX, N, SZVAL)

LIST1 (MAX) 整数型 何も入れなくてよい。
(n番目の要素がいかなる状態であるかが、入って戻ってくる。)

@ LIST2 (2*MAX) 整数型 何も入れなくてよい。
(いかなる状態が、何番目の要素であるか、が入ってもどってくる。)

@ MAX 整数型 行列の次元数
 $MAX = {}_N C_M$, $M = N/2 - S_{tot}^z$
($S_{tot}^z = m/2$, $m = 0, 2, 4, \dots$ N:even
 $m = 1, 3, 5, \dots$ N:odd)

@ N 整数型 スピンの総数 ($N \leq 20$)

@ SZVAL 倍精度実数型 N:even $SZVAL = S_{tot}^z$
($S_{tot}^z = m/2$, $m = 0, 2, 4, \dots$)
N:odd $SZVAL = S_{tot}^z - 1/2$
($S_{tot}^z = m/2$, $m = 1, 3, 5, \dots$)

(5) MRXELM; ハミルトニアン行列をあからさまに作る。

(行列次元数 $MAX \leq 1500$ について有効。それ以上では
東大センターの D-class job では領域不足で
実行されない。)

CALL MRXELM(MAX, IPAIR, IBOND, LIST1, LIST2, N, BONDWT, ELEMENT, ZRTIO)

@ MAX	整数型	行列の次元数 $MAX = \binom{N}{M}$, $M = N/2 - S_{tot}^z$ $(S_{tot}^z = m/2, m = 0, 2, 4, \dots N: \text{even}$ $m = 1, 3, 5, \dots N: \text{odd})$
@ IPAIR(2*IBOND)	整数型	相互作用している site のペア (ボンド) $(2n-1$ 番目と $2n$ 番目 ($n = 1, 2, 3, \dots$) の要素が相互作用しているように代入すること)
@ IBOND	整数型	ボンドの総数
@ LIST1(MAX)	整数型	n 番目の要素がいかなる状態であるかを、 入れておく。(→サブルーチン SZ で求める。)
@ LIST2(2*IBOND)	整数型	いかなる状態が、何番目の要素であるかを 入れておく。(→サブルーチン SZ で求める。)
@ N	整数型	スピンの総数 ($N \leq 20$)
@ BONDWT(IBOND)	倍精度実数型	相互作用 J_{ij} (代入の仕方は ZRTIO に同じ。)
# ELEMENT(MAX*(MAX+1)/2)	倍精度実数型	何も入れなくてよい。 (ハミルトニアン行列が入って戻って来る。)

この行列は、圧縮型であり、MSL II の
 DEF 4 M で対角化するような形式に
 なっている。なお、S-810/20上で
 走らせるときも、 DEF 4 M の名前のまま
 でよい。)

@ ZRTIO (IBOND) 倍精度実数型 異方性 Δ_{ij}
 (IPAIR の要素と同じ順序になるように、
 代入すること。つまり、ZRTIO(n)は、
 IPAIR(2n-1)-IPAIR(2n) の相互作用)

(6) ORTHG ;ベクトルの直交化

CALL ORTHG (EV, NORM, IDGN, MAX, NUMVEC)

@ EV (MAX, NUMVEC)	倍精度実数型	直交化すべきベクトル (入力)
#		規格直交化されたベクトル (出力)
# NORM (NUMVEC)	整数型	何もいれなくてよい。 (規格直交化されたベクトルのノルムが 入って戻ってくる。) (残ったものには、1 が、消えたものに 0 が、入っている。)
# IDGN	整数型	何も入れなくてよい。 (縮退度が入って戻ってくる)
@ MAX	整数型	行列の次元数
@ NUMVEC	整数型	直交化すべきベクトルの数

(7) CHECK ; 求まったベクトルが、しかるべき
固有ベクトルかどうか

CALL CHECK(X, V, LIST1, LIST2, MAX, N, IPAIR, IBOND, ZRTIO, BONDWT, PRD)

@ X (MAX)	倍精度実数型	CHECKすべきベクトル
# V (MAX)	倍精度実数型	何も入れなくてよい。 (Xにハミルトニアン行列をかけて、得られた ベクトルが、入って戻って来る。)
@ LIST1 (MAX)	整数型	n番目の要素がいかなる状態であるかを、 入れておく。(→サブルーチンSZで求める。)
@ LIST2 (2*MAX)	整数型	いかなる状態が、何番目の要素であるかを 入れておく。(→サブルーチンSZで求める。)
@ MAX	整数型	行列の次元数
@ N	整数型	スピンの総数 ($N \leq 20$)
@ IPAIR (2*IBOND)	整数型	相互作用している site のペア (ボンド) ($2n-1$ 番目と $2n$ 番目 ($n=1, 2, 3, \dots$) の要素が相互作用しているよう に代入すること)
@ IBOND	整数型	ボンドの総数
@ ZRTIO (IBOND)	倍精度実数型	異方性 Δ_{ij} (IPAIR の要素と同じ順序になるように、 代入すること。つまり、ZRTIO(n)は、 $IPAIR(2n-1)-IPAIR(2n)$ の相互作用)

田口善弘, 西森秀稔

@ BONDWT (IBOND) 倍精度実数型 相互作用 J_{ij}
(代入の仕方はZRTIO に同じ。)

PRD 倍精度実数型 固有値 $X^* H X$

Appendix. B

プログラム・パッケージ "TITPACK" をすぐ使う為に。

本プログラム・パッケージ TITPACKによって、基底状態とその付近のエネルギー固有値を得るためには、Fig. 1のような順序で、サブルーチンをHITAC S-810/20 上で実行する必要がある。Fig. 1の太線に沿って実行をする際のメイン・プログラム例をAppendix. Eに示した。参照しながら読んでほしい。なお、これらのサブルーチンのソース・リストは、Appendix. Dにある。また、東大センターのB5085. TITPACK. FORTにも入っており、すべてのuserに読出しをPERMITしてあるので、COPYして使って下さい。(86年4月以降は、B5085→A85085となる予定。)

I. メイン・ルーチンの作成

1次元リング状反強磁性 ($N=18$) の場合が、Appendix. Eにある。大抵の場合、このメイン・ルーチン例に多少、手を加えればよいはずである。以下では、どこをいじればよいかを述べておく。(_____ の部分)

①PARAMETER ($N=18$, $MAX=48620$, $IBOND=N$)

N; SPINの総数

MAX; 行列の次元 (決め方はAppendix. AのLANCZSの項等を見よ。)

IBOND; ボンドの総数

②DATA BONDWT/ $IBOND * (-1.0D0)$ /

③DATA ZRTIO/ $IBOND * 1.0D0$ /

④C***SITE INFORMATION

$IPAIR(1) = 1$

:

:

$10 \quad IPAIR(I) = I/2 + 1$

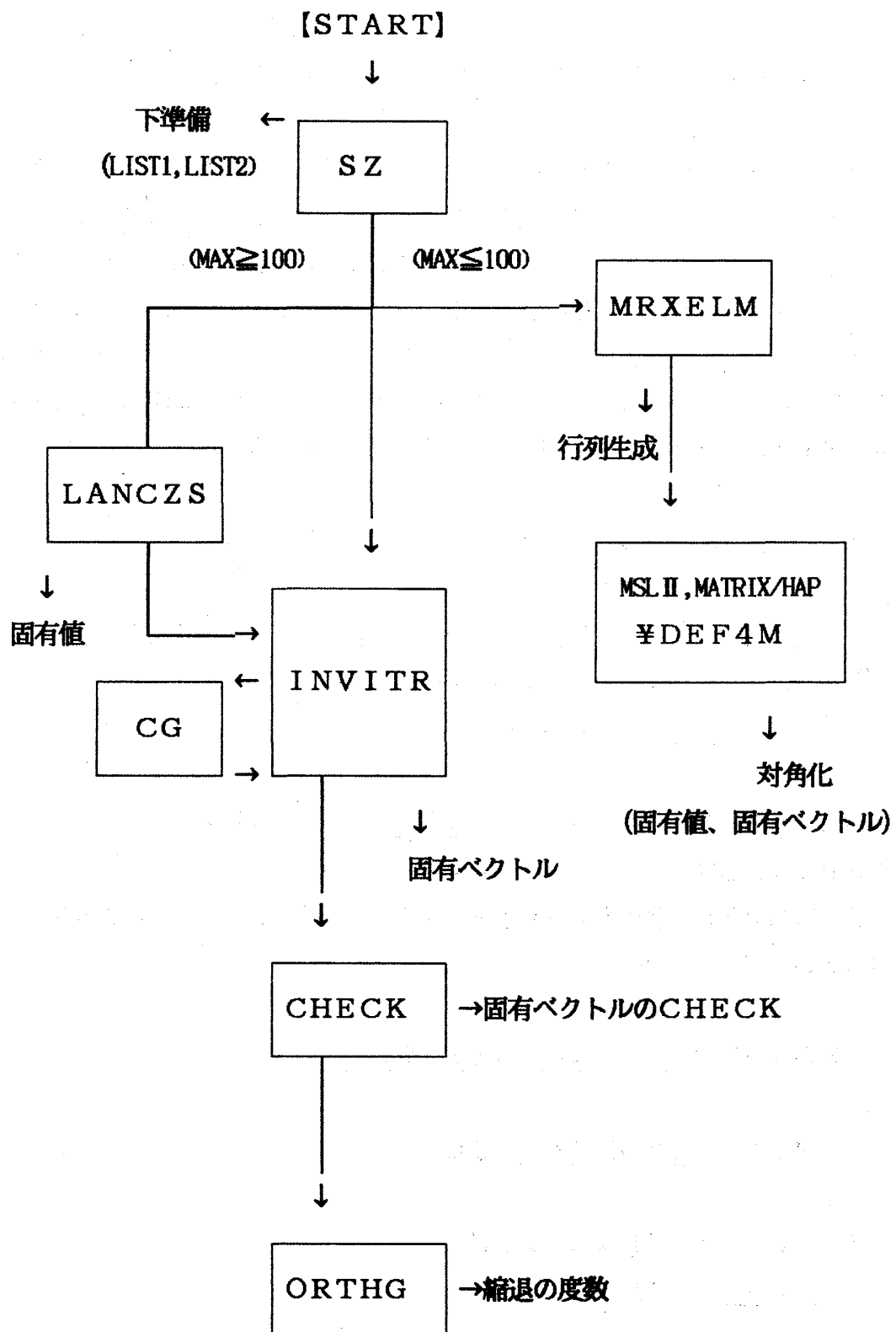


Fig. 1 TITPACK 流れ図

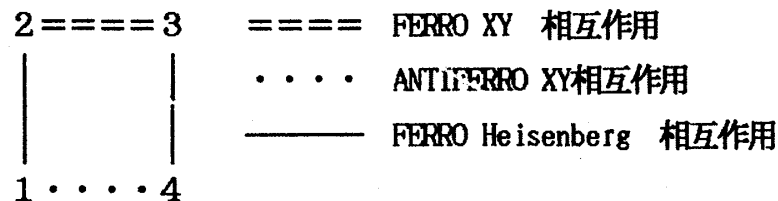
②～④は、組になっているので、一緒に述べる。②～④は、ハミルトニアンを指定する配列である。その為にはまず、対角化したいハミルトニアンを、下の形に書き直す必要がある。

$$\mathcal{H} = -2 \sum_{\langle i, j \rangle} J_{ij} \{ \Delta_{ij} S_i^z S_j^z + 1/2 (S_i^+ S_j^- + S_i^- S_j^+) \}$$

IPAIRは、 $\langle i, j \rangle$ を記憶する配列。BONDWTは、 J_{ij} を記憶する配列である。
ZRTIOは、 Δ_{ij} を記憶する配列である。メイン・ルーチン例では、

$$\mathcal{H} = 2 \sum_{i=1}^{18} S_i \cdot S_{i+1} \quad (\text{ただし、} S_{19} = S_1)$$

というハミルトニアンを対角化することになっているので、見て欲しい。具体的には、IPAIRには、 $2n-1$ 番目と $2n$ 番目 ($n=1, 2, \dots$) が相互作用するように、siteの番号を並べる。そして、それに合わせた順番で、BONDWT, ZRTIOに J_{ij} , Δ_{ij} を代入する。例えば、下図の様なときは、



```
DATA IPAIR /1, 2, 2, 3, 3, 4, 1, 4/
DATA BONDWT /1.0D0, 1.0D0, 1.0D0, -1.0D0/
DATA ZRTIO /1.0D0, 0.0D0, 1.0D0, 0.0D0/
となる。
```

⑤ CALL SZ (LIST1, LIST2, MAX, N, 0.0D0)

_____の部分は、SZVALと命名されているパラメーターである。(Appendix. A参照) 本プログラムでは、 S_{site} の固有空間ごとに対角化を実行し、その空間内の基底エネルギーを計算しているので、それを指定してやる必要がある。決め方は、マニュアルのSZの項を見よ。

以上、①～⑤の変更で、答えが得られる。ちなみに、基底エネルギーは、LANCZSのE(1)～E(4)に J_{ij} と同じ単位で出力される。すなわち、Appendix. Eの出力例で、

$E(1) = -16.04549817$ となっているのは、ハミルトニアンを

$$\mathcal{H} = 2J \sum_{i=1}^{18} S_i \cdot S_{i+1} \quad (\text{ただし, } S_{19} = S_1)$$

とした場合、基底状態のエネルギー $E_g/J = -16.04549817$ を意味する。また、直交化された固有ベクトルは ORTHG の $V(\text{MAX}, 3)$ という配列に出力されているので、必要な場合は、その都度、とり出せばよろしい。〔(3, 4) の縮退度が、3 と出た時は、このルーチンではダメである。というのは、3本のベクトルを直交化しているにすぎないので、縮退度が、4でも、5でも、100でも、3になってしまうからである。3になったときは、より多くのベクトルをとりだして直交化するように、メイン・ルーチンを書きかえる必要がある。〕

各固有ベクトルの意味は、以下の通りである。本サブルーチン・パックでは、波動関数の2進数表示を採用している。つまり、スピンの up-down 表示 \uparrow と \downarrow を、1 と 0 に置きかえて並べ、2進数として読むのである。例えば、 $N=4$ で "2" というとき、0010、つまり、 $\downarrow\downarrow\uparrow\downarrow$ という状態を表現しており、site 1, 3, 4, が down, site 2 が up である。 $N=12$ で 948 というとき、001011011100、これは、 $\downarrow\downarrow\uparrow\downarrow\uparrow\uparrow\downarrow\uparrow\uparrow\downarrow\downarrow$ に相当し、site 3, 4, 5, 7, 8, 10, が up, site 1, 2, 6, 9, 11, 12, が down という状態になる。具体的に、 $V(\text{MAX}, 3)$ の配列の i 番目の要素が示す基底状態の波動関数が何であるかは、LIST1 (MAX) という配列の i 番目に入っている。例えば、 $N=4$ 、 $S_{\text{tot}}^z=0$ では、 $\text{MAX}=6$ であり、LIST1 (MAX) という配列には、3, 5, 6, 9, 10, 12, が入っている。これが、意味するところは、 $V(\text{MAX}, 3)$ という波動関数は、

$$\begin{aligned} V(\text{MAX}, I) = & V(1, I) |3\rangle + V(2, I) |5\rangle \\ & + V(3, I) |6\rangle + V(4, I) |9\rangle \\ & + V(5, I) |10\rangle + V(6, I) |12\rangle \\ & (I=1, 2, 3) \end{aligned}$$

という波動状態を示しているということである。より詳しくは、本文の解説部分、§2、§4、を参照のこと。

(注) LANCZOS 法は、 $\text{MAX} \leq 100$ の時には、経験によると、ちゃんとは走ってくれない。そこで、 $\text{MAX} \leq 100$ の時は、LANCZS, INVITR をいっさいとりやめて、

MAXELMを呼び、ハミルトニアン行列を作り、HITACのライブラリー・プログラムであるMSL IIや、MATRIX/HAPで、対角化する必要がある。この時は、メインルーチン例の、「C*** EIGENVALUES」以下の行を、全て取り去り、代わりに、「CALL MRXELM(.....)」とやって、得られた1次元配列、ELEMNTを、MSL IIのDEF 4 Mで対角化する。そのメイン・ルーチン例も、Appendix. Fにつけておいた。(S-810/20上で走らせるときも、DEF 4 Mを呼べばよい。自動的に、MATRIX/HAPが起動する。) なお、メインルーチン例は、このままでは、 $N=20$ では走ってくれない。というのは、ORTHGのところで、直交化する為に、V (MAX, 3) などというぜいたくなことをしているためである。よって、 $N=20$ では、ORTHGをとりやめて、V (MAX, 3) という配列宣言をせずに走らせ、INVITRの出力のX (MAX) をデータセットに取り出しておいて後から直交化してみるという風にする必要がある。

II. COMPILE AND GO

これを 東大センターのHITAC S-810/20で走らせるには、以下のようにする。ソース・プログラム (メイン・ルーチン) はSOURCE. FORT: SPINに入っているとする。

(_____ に注意)

```
LOGON B9999/FOOL S(1500)
:
:
>>USE SOURCE. FORT @SPIN. LOAD HAP
>>COM SPIN, ,PA (COMARY)
:
    いろいろなメッセージ。気にする必要なし。
:
>>
```

以上の様な手続きでメインルーチンとサブルーチン・ライブラリーのロード・モジュールを作成する。メイン・ルーチンのロード・モジュールは@SPIN. LOAD, サブ・ルーチンのロード・モジュールは@SUB. LOADに入っていたとする。(なお、B5085. TIT PACK. FORTにあるデータセットには、ソース・リストしか与えられていないので、上

の手続きでCOMPILEしておかないといけない。) 実行には、JOB文を作らなければならない。

```
>>USE JCL.CNTL
>>E A,NEW                      (必要に応じてB~Dに変える。注1)
INPUT                            ↓
100: //B9999A JOB ***** ,CLASS=A, NOTIFY=B9999
200: //**MAIN SYSTEM=S810, DEST=T
300: >>USE ,@SPIN. LOAD
400: >>MATRIX
500: >>GO MAIN, LIB(@SUB. LOAD)
END S
:
:
>>SUB JCL.CNTL:A
```

注1) A~Cは、最大記憶領域8MBのジョブで、CPU-TIMEの長さが異なる。Dは、16MBのジョブ。よってMAXの値によって、8MBのジョブにするか、16MBのジョブにするかに気をつけなくてはならない。その決めかたを以下にごく簡単に述べておく。それには、メインルーチンで配列宣言されている主な配列の占有する領域を評価して見ればよい。整数型は配列の要素1ヶあたり4バイト。倍精度実数型は、配列の要素1ヶあたり8バイト。例えば、メインルーチン例では、

```
V (MAX, 3) .....  $3 \times \text{MAX} \times 8 = 24 \text{MAX}$  バイト
LIST1 (MAX) .....  $4 \times \text{MAX} = 4 \text{MAX}$  バイト
LIST2 (2**N) .....  $4 \times 2^N = 4 \times 2^{18}$  バイト
V0 (MAX), V1 (MAX), V2 (MAX), DELM (MAX),
B (MAX), X (MAX) ..... 全部で  $8 \times \text{MAX} \times 6 = 48 \text{MAX}$  バイト
```

よって、 $(24 + 4 + 48) \times \text{MAX} + 4 \times 2^{18} = 4.743696 \text{ MB}$
 \parallel
48620

こうやって計算した占有領域が8MB以下なら、A~Cジョブ、8MB以上なら、Dジョブを用いる。また、計算時間の大体の見積も述べておく。CPU-TIMEは、MAXとIBONDの積に比例する。Appendix. Eのプログラムで、1分14秒というところであるから、 $N=20$, $S_{\text{tot}}^z=0$ の同じモデル(1次元リング状)を解いたとすると、 $MAX=184756$ 、 $IBOND=20$ となるので、まあ、5分強というところであろうか。

最後に一言。Appendix. Eの出力例では、INVITR, CG, CHECKからメッセージが出されているが、Appendix. D及びB5085. TITPACK. FOR Tにあるソース・リストではこのようなメッセージが出ないようにわざとコメント文にしてある。必要に応じて、"C" をとりさり、メッセージを出力させるとよい。

※コメント文になっているWRITE文の位置

CG.....「150 FORMAT」の上

INVITR.....「100 FORMAT」の上

CHECK.....「100, 110, 111 FORMAT」の上

Appendix. C

ERROR メッセージ 解説

☆……解説

★……対策

(A) from SZ

TOO LARGE MAX OR N GIVEN TO SZ

☆行列の次元数 (MAX) か、スピンの
総数 (N) が大きすぎる。

($N \leq 20, \text{MAX} \leq 184756$)

★あきらめる。

SZVAL SHOULD NOT BE NEGATIVE

☆仮引数SZVAL が負である。

★SZVAL の値を確かめる。

SZVAL SHOULD NOT EXCEED N/2

☆仮引数SZVAL が $N/2$ より大きい。

★SZVAL の値が $N/2$ を越えることは、
物理的におかしい。(Appendix. A
SZの項を見よ。) 値をCHECKする。

INCORRECT MAX OR N GIVEN TO SZ

☆SZVAL に対してMAX かNの与え方が、
間違っている。

★MAX, N, SZVALには、一定の関係がある。
(Appendix. A, SZの項を見よ。)
MAX, N, SZVALが、
この関係を満たしているかどうかを、
確かめる。

(B) from LANCZS

INCORRECT IV GIVEN TO LANCZS

☆初期ベクトルを指定するパラメータIVの値が
間違っている。

★ $1 \leq IV \leq \text{MAX}$ であるかどうか確かめる。

TOO LARGE MAX OR N GIVEN TO LANCZS

☆行列の次元数 (MAX) か、スピンの総数 (N) が、大きすぎる。

★ $N \leq 20, \text{MAX} \leq 184756$, を満たしているかどうか確かめる。

INCORECT DATA IN IPAIR FOUND
IN LANCZS

☆IPAIR の内容がおかしい。

★IPAIR の内容を確認する。特に、ゼロになっていないか? (データの数が、 $\text{IBOND} \times 2$ より少なくないか?)

TRIDIAGONALIZATION UNSUCCESSFUL
IN LANCZS

☆三重対角化ができない。

★初期ベクトルを変えて試みる。
($\text{MAX} \leq 100$ だとどんな初期ベクトルでもうまくいかないことが多い。)

LANCZS DID NOT CONVERGE

☆LANCZOS法が、収束しない。

★あきらめる (or 反復回数を、
(DO 100 I=2, 500の500) 増やしてみる。)

(C) from INVITR

TOO LARGE MAX OR N GIVEN TO INVITR

☆行列の次元数 (MAX) か、スピンの総数 (N) が、大きすぎる。

★ $N \leq 20, \text{MAX} \leq 184756$, を満たしているかどうか確かめる。

ITERAT IN CG IS GT MAX
ITR IN INVITR IS **

☆CG法は、行列の次元程度の反復回数で収束するはずなのに、反復回数が、それ以上になってしまったので、打ち切った。そのため逆反復法が**回までしか反復できなかった。

★あきらめる (or CG法の反復回数を、
(サブルーチン CGの DO 20 ITERAT=1, MAX のMAX) 増やす。)

INVERSE ITERATION DID NOT CONVERGE

☆逆反復法が収束しない。

★あきらめる。(or (1) 反復回数を
(DO 20 ITR=1,20の20) 増やす。
(2) INVITRに与える近似固有値の精度
を上げる。)

(D) from CG

INCORECT DATA IN IPAIR FOUND IN CG

☆IPAIR の内容がおかしい。

★IPAIR の内容を確認する。特に、ゼロに
なっていないか? (データの数、IBOND*2
より少なくないか?)

CG DID NOT CONVERGE

☆CG法が収束しない。

★あきらめる(or 反復回数を、
(DO 20 ITERAT=1,MAX のMAX) 増やす。)

(E) from ORTHG

NUMBER OF VECTORS LESS THAN 2
IN ORTHG

☆直交化すべきベクトルが1本しか、与
えられていないので、何もしない。

★直交化すべきベクトルを2本以上与えるよう
にするか、ORTHGを呼ぶのをやめる。

NULL VECTOR GIVEN TO ORTHG.
LOCATION IS *

☆ゼロ・ベクトルが直交化すべきベクトルの中
に入っている。それは、*番目のベクトルで
ある。

★ゼロ・ベクトルを除いてから、ORTHGを
呼ぶようにする。

NON-ORTHOGONAL VECTORS AT * **

☆*番目のベクトルと**番目のベクトルが
誤差の積み重なりで完全には直交化できない。

★近似的には (10^{-10} より悪い精度で) 直交化
できているのでそれで満足するか、あるいは
もう1度 ORTHG を通す。ただし、後者の
場合固有ベクトルであるという性質が誤差の
重なるため、失われることがある。
サブルーチン CHECK を使って確かめた
方がよい。

(F) from MRXELM

INCORRECT DATA IN IPAIR FOUND
IN MRXELM

☆IPAIR の内容がおかしい。
★IPAIR の内容を確認する。特に、ゼロ
になっていないか? (データの数が、
IBOND*2 より少なくないか?)

Appendix. D TITPACK のソースリスト

```

00010 C***** CONFIGURATIONS WITH SPECIFIED SZ *****
00020
00030     SUBROUTINE SZ(LIST1,LIST2,MAX,N,SZVAL)
00040
00050 C*** VARIABLES MARKED @ SHOULD BE GIVEN FROM MAIN PROGRAM
00060 C*** VARIABLES MARKED # ARE EVALUATED AND RETURNED
00070 C   LIST1(I) # I-TH SPIN CONFIGURATION EXPRESSED BY BIT PATTERN
00080 C   LIST2     # INVERSE LIST OF LIST1
00090 C   MAX      @ DIMENSION OF THE MATRIX
00100 C   N        @ NUMBER OF SITES OF THE LATTICE
00110 C   SZVAL    @ TOTAL SZ OF THE SPACE TO BE DIAGONALIZED
00120
00130     REAL*8 SZVAL
00140     DIMENSION LIST1(MAX),LIST2(2**N)
00150
00160     IF(MAX.GT.184756.OR.N.GT.20)THEN
00170       WRITE(6,*)' TOO LARGE MAX OR N GIVEN TO SZ'
00180       STOP
00190     END IF
00200     IF(SZVAL.LT.-1.0D-20)THEN
00210       WRITE(6,*)' SZVAL SHOULD NOT BE NEGATIVE'
00220       STOP
00230     END IF
00240     IF(SZVAL.GT.N/2.D0+1.D-20)THEN
00250       WRITE(6,*)' SZVAL SHOULD NOT EXCEED N/2'
00260       STOP
00270     END IF
00280     IUPSPN=N/2+MOD(N,2)+INT(SZVAL+0.001D0)
00290     ICNT=0
00300     DO 10 I=1,2**N
00310       ISZ=0
00320       LIST2(I)=0
00330       DO 20 J=0,N-1
00340         ISZ=ISZ+MOD(I/2**J,2)
00350         IF(ISZ.NE.IUPSPN)GO TO 10
00360         ICNT=ICNT+1
00370         IF(ICNT.GT.MAX)THEN
00380           WRITE(6,*)' INCORRECT MAX OR N GIVEN TO SZ'
00390           STOP
00400         END IF
00410         LIST1(ICNT)=I
00420         LIST2(I)=ICNT
00430       10 CONTINUE
00440       IF(ICNT.EQ.MAX)RETURN
00450       WRITE(6,*)' INCORRECT MAX OR N GIVEN TO SZ'
00460       STOP
00470     END
00480
00490 C***** DIAGONALIZATION BY THE LANZOS-BISECTION ALGORITHM *****
00500
00510     SUBROUTINE LANZOS(N,IPAIR,IBOND,V2,V1,V0,DELM,ZRTIO,
00520     & ISTEP,E,I,IV,LIST1,LIST2,MAX,BONDWT)
00530
00540 C*** VARIABLES MARKED @ SHOULD BE GIVEN FROM MAIN PROGRAM
00550 C*** VARIABLES MARKED # ARE EVALUATED AND RETURNED
00560 C   N        @ NUMBER OF SITES
00570 C   IPAIR(IBOND*2) @ PAIRS OF SITES CONNECTED BY BONDS
00580 C   IBOND    @ NUMBER OF BONDS
00590 C   V0,V1,V2 WORKING AREA
00600 C   DELM(MAX) WORKING AREA FOR DIAGONAL ELEMENTS OF HAMILTONIAN

```

```

00610 C      ZRTIO(IBOND)      @ RATIO OF Jz TO Jxy
00620 C      ISTEP            @ INTERVAL TO CHECK CONVERGENCE
00630 C      E(4)             # FOUR LOWEST EIGENVALUES TO BE RETURNED
00640 C      I                # NUMBER OF ITERATIONS TO BE RETURNED
00650 C      IV              @ LOCATION OF NONZERO ELEMENT OF THE INITIAL VECTOR
00660 C      LIST1,LIST2      @ CONFIGURATIONS IN THE SPACE OF THE SPECIFIED SZ
00670 C      MAX              @ DIMENSION OF THE MATRIX
00680 C      BONDWT(IBOND)    @ EXCHANGE INTERACTION OF EACH BOND Jxy(IJ)
00690
00700 C
00710 C      OPTIMIZED FOR S-810/20
00720 C      1985/2/22 ; REVISED ON 1985/10/21
00730
00740      IMPLICIT REAL*8(A-H,O-Z)
00750      INTEGER SITE1,SITE2
00760      DIMENSION E(4),IPAIR(IBOND*2),BONDWT(IBOND),ZRTIO(IBOND)
00770      DIMENSION LIST1(MAX),LIST2(2**N)
00780      DIMENSION VO(MAX),V1(MAX),V2(MAX),DELM(MAX)
00790      DIMENSION ALPHA(500),BETA(500),WK(500,4),IWK(500)
00800
00810      IF(IV.LE.0.OR.IV.GT.MAX)THEN
00820          WRITE(6,*)' INCORRECT IV GIVEN TO LANCZS'
00830          RETURN
00840      END IF
00850      IF(MAX.GT.184756.OR.N.GT.20)THEN
00860          WRITE(6,*)' TOO LARGE MAX OR N GIVEN TO LANCZS'
00870          STOP
00880      END IF
00890 C*** INITIALIZATION
00900      DO 10 I=1,MAX
00910          VO(I)=0.0D0
00920          V1(I)=0.0D0
00930          DELM(I)=0.0D0
00940 10    CONTINUE
00950
00960 C*** DIAGONAL ELEMENTS
00970      DO 20 I=1,IBOND
00980          SITE1=IPAIR(I*2-1)-1
00990          SITE2=IPAIR(I*2 )-1
01000          IS1=2**SITE1
01010          IS2=2**SITE2
01020          IF(IS1.LE.0.OR.IS2.LE.0)THEN
01030              WRITE(6,*)' INCORRECT DATA IN IPAIR FOUND IN LANCZS'
01040              STOP
01050          END IF
01060          IS=IS1+IS2
01070          WGHT=BONDWT(I)*ZRTIO(I)
01080          DO 20 J=1,MAX
01090              IBIT=IAND(LIST1(J),IS1)+IAND(LIST1(J),IS2)
01100              IF(IBIT.EQ.0.OR.IBIT.EQ.IS)THEN
01110                  PARREL= 0.5D0
01120              ELSE
01130                  PARREL=-0.5D0
01140              END IF
01150              DELM(J)=DELM(J)-WGHT*PARREL
01160 20    CONTINUE
01170      VO(IV)=DELM(IV)
01180      V1(IV)=1.0D0
01190
01200 C*** ALPHA(1) AND BETA(1)

```



```

01210      ALPHA(1)=DELM(IV)
01220      DO 30 K=1,IBOND
01230          SITE1=IPAIR(K*2-1)-1
01240          SITE2=IPAIR(K*2 )-1
01250          IS1=2**SITE1
01260          IS2=2**SITE2
01270          IS=IS1+IS2
01280          WGHT=BONDWT(K)
01290          DO 30 I=1,MAX
01300              IBIT=IAND(LIST1(I),IS1)+IAND(LIST1(I),IS2)
01310              IF(IBIT.EQ.0.OR.IBIT.EQ.IS)GO TO 30
01320              IEXCHG=IEOR(IEOR(LIST1(I),IS1),IS2)
01330              VO(I)=VO(I)-V1(LIST2(IEXCHG))*WGHT
01340 30      CONTINUE
01350      VO(IV)=VO(IV)-ALPHA(1)
01360      BETA1=0.0D0
01370      DO 50 I=1,MAX
01380 50      BETA1=BETA1+VO(I)**2
01390      BETA(1)=SQRT(BETA1)
01400      IF(BETA(1).LT.0.5D-20)THEN
01410          BETA(1)=0.0D0
01420          DO 60 I=1,MAX
01430 60      V2(I)=0.0D0
01440          IDX=IV+1
01450          IF(IDX.GT.MAX)IDX=1
01460          V2(IDX)=1.0D0
01470      ELSE
01480          DO 65 I=1,MAX
01490 65      V2(I)=VO(I)/BETA(1)
01500      END IF
01510
01520 C*** ITERATION
01530      DO 100 I=2,500
01540          DO 105 J=1,MAX
01550 105      VO(J)=DELM(J)*V2(J)
01560          DO 110 K=1,IBOND
01570              SITE1=IPAIR(K*2-1)-1
01580              SITE2=IPAIR(K*2 )-1
01590              IS1=2**SITE1
01600              IS2=2**SITE2
01610              IS=IS1+IS2
01620              WGHT=BONDWT(K)
01630              DO 110 J=1,MAX
01640                  IBIT=IAND(LIST1(J),IS1)+IAND(LIST1(J),IS2)
01650                  IF(IBIT.EQ.0.OR.IBIT.EQ.IS)GO TO 110
01660                  IEXCHG=IEOR(IEOR(LIST1(J),IS1),IS2)
01670                  VO(J)=VO(J)-V2(LIST2(IEXCHG))*WGHT
01680 110      CONTINUE
01690      ALPHAI=0.0D0
01700      DO 120 J=1,MAX
01710 120      ALPHAI=ALPHAI+V2(J)*VO(J)
01720      ALPHA(I)=ALPHAI
01730      BETAI1=BETA(I-1)
01740      DO 130 J=1,MAX
01750 130      VO(J)=VO(J)-ALPHAI*V2(J)-BETAI1*V1(J)
01760      BETAI=0.0D0
01770      DO 140 J=1,MAX
01780          V1(J)=V2(J)
01790          BETAI=BETAI+VO(J)**2
01800 140      CONTINUE

```

```

01810      BETA(I)=SQRT(BETAI)
01820      IF(BETA(I).LT.0.5D-20)THEN
01830        WRITE(6,*)' TRIDIAGONALIZATION UNSUCCESSFUL IN LANCZS'
01840        RETURN
01850      END IF
01860      IF(I.GT.20.AND.MOD(I,ISTEP).EQ.0)THEN
01870        CALL %DET2M(ALPHA,I,BETA,4,1,0.5D-9,4,E,V2,I,IFLG,
01880      &          WK,IWK,IER)
01890      &      IF(ABS((EBEFOR-E(2))/E(2)).LT.1.0D-7)RETURN
01900        EBEFOR=E(2)
01910      END IF
01920      DINVBT=1.0D0/BETA(I)
01930      DO 150 J=1,MAX
01940 150      V2(J)=V0(J)*DINVBT
01950      IF(I.EQ.20)THEN
01960        CALL %DET2M(ALPHA,I,BETA,20,1,0.5D-9,4,
01970      %          E,V0,20,IFLG,WK,IWK,IER)
01980        EBEFOR=E(2)
01990      END IF
02000 100      CONTINUE
02010      WRITE(6,*)' LANCZS DID NOT CONVERGE'
02020      RETURN
02030      END
02040
02050 C***** MATRIX ELEMENTS FOR GENERAL BOND WEIGHTS *****
02060
02070      SUBROUTINE MRXELM(MAX,IPAIR,IBOND,LIST1,LIST2,N,BONDWT,ELEMNT,
02080      &          ZRTIO)
02090
02100 C*** VARIABLES MARKED @ SHOULD BE GIVEN FROM MAIN PROGRAM
02110 C*** VARIABLES MARKED # ARE EVALUATED AND RETURNED
02120 C      MAX      @ MATRIX DIMENSION
02130 C      IPAIR     @ PAIRS OF SITES CONNECTED BY BONDS
02140 C      IBOND     @ NUMBER OF BONDS
02150 C      LIST1,2   @ CONFIGURATIONS IN THE SPACE OF THE SPECIFIED SZ
02160 C      N         @ NUMBER OF SITES
02170 C      BONDWT    @ EXCHANGE INTERACTION OF EACH BOND Jxy(IJ)
02180 C      ELEMNT    # MATRIX ELEMENTS TO BE RETURNED
02190 C      ZRTIO     @ RATIO OF Jz TO Jxy
02200
02210      IMPLICIT REAL*8 (A-H,O-Z)
02220      INTEGER SITE1,SITE2
02230      DIMENSION IPAIR(IBOND*2),ELEMNT(MAX*(MAX+1)/2),BONDWT(IBOND)
02240      DIMENSION LIST1(MAX),LIST2(2**N),ZRTIO(IBOND)
02250
02260 C*** INITIALIZATION
02270      DO 10 I=1,MAX*(MAX+1)/2
02280 10      ELEMNT(I)=0.0D0
02290
02300 C*** ELEMENTS
02310      DO 30 K=1,IBOND
02320        SITE1=IPAIR(K*2-1)-1
02330        SITE2=IPAIR(K*2 )-1
02340        IF(SITE1.LE.-1.OR.SITE1.LE.-1)THEN
02350          WRITE(6,*)' INCORRECT DATA IN IPAIR FOUND IN MRXELM'
02360          STOP
02370        END IF
02380        IS1=2**SITE1
02390        IS2=2**SITE2
02400        IS=IS1+IS2

```

```

02410          WGHT=BONDWT(K)
02420          DIAG=WGHT*0.5D0*ZRTIO(K)
02430 *VOPTION VEC
02440          DO 30 I=1,MAX
02450              IBIT=IAND(LIST1(I),IS1)+IAND(LIST1(I),IS2)
02460              IDG=I*(I+1)/2
02470              IF(IBIT.EQ.0.OR.IBIT.EQ.IS)THEN
02480                  ELEMNT(IDG)=ELEMNT(IDG)-DIAG
02490              ELSE
02500                  ELEMNT(IDG)=ELEMNT(IDG)+DIAG
02510              IEXCHG=LIST2(IEOR(IEOR(LIST1(I),IS1),IS2))
02520              IF(IEXCHG.GE.I)GO TO 30
02530              ELEMNT(I*(I-1)/2+IEXCHG)=-WGHT
02540          END IF
02550 30      CONTINUE
02560          RETURN
02570          END
02580
02590 C***** INVERSE ITERATION *****
02600
02610          SUBROUTINE INVITR(E,IPAIR,IBOND,LIST1,LIST2,MAX,N,
02620          &              B,X,P,R,Y,ISTEP,IV,ZRTIO,BONDWT)
02630 C*** VARIABLES MARKED @ SHOULD BE GIVEN FROM MAIN PROGRAM
02640 C*** VARIABLES MARKED # ARE EVALUATED AND RETURNED
02650 C      E          @ APPROXIMATE EIGENVALUE
02660 C      IPAIR      @ PAIRS OF SITES CONNECTED BY BONDS
02670 C      IBOND      @ NUMBER OF BONDS
02680 C      LIST1,2    @ CONFIGURATIONS IN THE SPACE OF THE SPECIFIED SZ
02690 C      MAX        @ MATRIX DIMENSION
02700 C      N          @ NUMBER OF SITES
02710 C      B          WORKING ARE FOR RHS OF THE EQUATION (H-E(APPROX))*X=B
02720 C      X          # EIGEN VECTOR TO BE RETURNED
02730 C      P,R,Y      WORKING AREA USED IN THE CG ROUTINE
02740 C      ISTEP      @ INTERVAL TO CHECK CONVERGENCE IN CG
02750 C      IV         @ LOCATION OF THE NONZERO ELEMENT IN THE INITIAL VECTOR
02760 C      ZRTIO      @ RATIO OF Jz TO Jxy
02770 C      BONDWT    @ EXCHANGE INTERACTION OF EACH BOND Jxy(ij)
02780
02790          IMPLICIT REAL*8(A-H,O-Z)
02800          DIMENSION IPAIR(IBOND*2),ZRTIO(IBOND),LIST1(MAX),LIST2(2**N)
02810          DIMENSION B(MAX),X(MAX),R(MAX),Y(MAX),P(MAX),BONDWT(IBOND)
02820
02830          IF(MAX.GT.184756.OR.N.GT.20)THEN
02840              WRITE(6,*)' TOO LARGE MAX OR N GIVEN TO INVITR'
02850              RETURN
02860          END IF
02870          DO 10 I=1,MAX
02880 10      B(I)=0.0D0
02890          B(IV)=1.0D0
02900          DO 20 ITR=1,20
02910              CALL CG(E,IPAIR,IBOND,LIST1,LIST2,MAX,N,B,X,P,R,Y,ISTEP,ITERAT,
02920          &              ZRTIO,BONDWT)
02930              IF(ITERAT.GT.MAX)THEN
02940                  WRITE(6,*)' ITERAT IN CG IS GT MAX'
02950                  WRITE(6,*)' ITR IN INVITR IS',ITR
02960                  RETURN
02970              END IF
02980              XNORM=0.0D0
02990              DO 30 I=1,MAX
03000 30      XNORM=XNORM+X(I)**2

```

```

03010      XNORM=SQRT(XNORM)
03020      DO 40 I=1,MAX
03030  40      X(I)=X(I)/XNORM
03040      XB=0.0D0
03050      DO 50 I=1,MAX
03060  50      XB=XB+X(I)*B(I)
03070      IF(ABS(ABS(XB)-1.0D0).LT.1.0D-12)THEN
03080  C      WRITE(6,100)ITR
03090  100      FORMAT('          NUMBER OF ITERATIONS IN INVITR :',I5)
03100      RETURN
03110      END IF
03120      DO 60 I=1,MAX
03130  60      B(I)=X(I)
03140  20      CONTINUE
03150      WRITE(6,*)' INVERSE ITERATION DID NOT CONVERGE'
03160      RETURN
03170      END
03180
03190  C***** SOLUTION OF LINEAR EQUATIONS -- CG METHOD *****
03200
03210      SUBROUTINE CG(E,IPAIR,IBOND,LIST1,LIST2,MAX,N,
03220      &          B,X,P,R,Y,ISTEP,ITERAT,ZRTIO,BONDWT)
03230
03240  C*** VARIABLES MARKED @ SHOULD BE GIVEN FROM INVITR ROUTINE
03250  C*** VARIABLES MARKED # ARE EVALUATED AND RETURNED
03260  C      E          @ APPROXIMATE EIGENVALUE
03270  C      IPAIR      @ PAIRS OF SITES CONNECTED BY BONDS
03280  C      IBOND      @ NUMBER OF BONDS
03290  C      LIST1,2    @ CONFIGURATIONS IN THE SPACE OF SPECIFIED SZ
03300  C      MAX        @ MATRIX DIMENSION
03310  C      N          @ NUMBER OF SITES
03320  C      B(MAX)     @ RIGHT HAND SIDE OF THE EQUATION : (H-E(APPROX))*X=B
03330  C      X(MAX)     # SOLUTION
03340  C      P,R,Y      WORKING AREA
03350  C      ISTEP      @ INTERVAL TO CHECK CONVERGENCE
03360  C      ITERAT     # NUMBER OF ITERATIONS TO BE RETURNED
03370  C      ZRTIO      @ RATIO OF Jz TO Jxy
03380  C      BONDWT     @ EXCHANGE INTERACTION OF EACH BOND Jxy(ij)
03390
03400      IMPLICIT REAL*8(A-H,O-Z)
03410      INTEGER SITE1,SITE2
03420      DIMENSION IPAIR(IBOND*2),ZRTIO(IBOND),LIST1(MAX),LIST2(2**N)
03430      DIMENSION B(MAX),X(MAX),R(MAX),Y(MAX),P(MAX),BONDWT(IBOND)
03440
03450  C*** INITIALIZATION
03460      BNORM=0.0D0
03470      DO 10 I=1,MAX
03480          BNORM=BNORM+B(I)**2
03490          R(I)=B(I)
03500          P(I)=B(I)
03510          X(I)=0.0D0
03520  10      CONTINUE
03530
03540  C*** ITERATION
03550      DO 20 ITERAT=1,MAX
03560          DO 30 I=1,MAX
03570  30          Y(I)=0.0D0
03580          DO 40 K=1,IBOND
03590              SITE1=IPAIR(K*2-1)-1
03600              SITE2=IPAIR(K*2 )-1

```

```

03610      IS1=2**SITE1
03620      IS2=2**SITE2
03630      IF(IS1.LE.0.OR.IS2.LE.0)THEN
03640          WRITE(6,*)' INCORRECT DATA IN IPAIR FOUND IN CG'
03650          STOP
03660      END IF
03670      IS=IS1+IS2
03680      EPERBD=E/FLOAT(IBOND)
03690      WGHT=BONDWT(K)
03700      DIAG1= WGHT*0.5D0*ZRTIO(K)+EPERBD
03710      DIAG2=-WGHT*0.5D0*ZRTIO(K)+EPERBD
03720      DO 40 I=1,MAX
03730          IBIT=IAND(LIST1(I),IS1)+IAND(LIST1(I),IS2)
03740          IF(IBIT.EQ.0.OR.IBIT.EQ.IS)THEN
03750              Y(I)=Y(I)-DIAG1*P(I)
03760              ELSE
03770                  Y(I)=Y(I)-DIAG2*P(I)
03780                  Y(I)=Y(I)-
03790      &          P(LIST2(IEOR(IEOR(LIST1(I),IS1),IS2))) *WGHT
03800      END IF
03810 40      CONTINUE
03820      RP=0.0D0
03830      YP=0.0D0
03840      DO 50 I=1,MAX
03850          RP=RP+R(I)*P(I)
03860          YP=YP+Y(I)*P(I)
03870 50      CONTINUE
03880      ALPHA=RP/YP
03890      RNORM=0.0D0
03900      DO 60 I=1,MAX
03910          X(I)=X(I)+ALPHA*P(I)
03920          RNORM=RNORM+R(I)**2
03930 60      CONTINUE
03940      RNORM2=0.0D0
03950      DO 70 I=1,MAX
03960          R(I)=R(I)-ALPHA*Y(I)
03970          RNORM2=RNORM2+R(I)**2
03980 70      CONTINUE
03990      BETA=RNORM2/RNORM
04000      DO 90 I=1,MAX
04010 90      P(I)=R(I)+BETA*P(I)
04020      IF(MOD(ITERAT,ISTEP).NE.0)GO TO 20
04030      IF(SQRT(RNORM2).LT.0.5D-10*SQRT(BNORM))THEN
04040  C          WRITE(6,150)ITERAT
04050 150      FORMAT('          NUMBER OF ITERATIONS IN CG      :',I5)
04060      RETURN
04070      END IF
04080 20      CONTINUE
04090      WRITE(6,*)' CG DID NOT CONVERGE'
04100      RETURN
04110      END
04120
04130 C***** SCHMIDT ORTHOGONALIZATION OF THE EIGENVECTORS *****
04140
04150      SUBROUTINE ORTHG(EV,NORM,IDGN,MAX,NUMVEC)
04160
04170 C*** VARIABLES MARKED @ SHOULD BE GIVEN FROM MAIN PROGRAM
04180 C*** VARIABLES MARKED # ARE EVALUATED AND RETURNED
04190  C      EV      @# VECTORS TO BE ORTHOGONALIZED / ORTHOGONALIZED VECTORS
04200  C      NORM(J) #      NORM OF THE J-TH VECTOR RETURNED

```

```

04210 C   IDGN      #   DEGREE OF DEGENERACY
04220 C   NUMVEC    @   NUMBER OF VECTORS TO BE CHECKED
04230
04240       IMPLICIT REAL*8(A-H,O-Z)
04250       DIMENSION EV(MAX,NUMVEC),NORM(NUMVEC)
04260
04270       IF(NUMVEC.LE.1)THEN
04280         WRITE(6,*)' NUMBER OF VECTORS LESS THAN 2 IN ORTHG'
04290         RETURN
04300       END IF
04310       DO 10 I=1,NUMVEC
04320         DNORM=0.0D0
04330         DO 20 J=1,MAX
04340           20  DNORM=DNORM+EV(J,I)**2
04350           IF(DNORM.LT.1.0D-20)THEN
04360             WRITE(6,*)' NULL VECTOR GIVEN TO ORTHG. LOCATION IS',I
04370             RETURN
04380           END IF
04390           DNORM=1.0D0/SQRT(DNORM)
04400           DO 25 J=1,MAX
04410             25  EV(J,I)=EV(J,I)*DNORM
04420           10  CONTINUE
04430           IDGN=NUMVEC
04440           NORM(1)=1
04450
04460 C*** ORTHOGONALIZATION
04470       DO 30 I=2,NUMVEC
04480         NORM(I)=1
04490         DO 40 J=1,I-1
04500           PRJCT=0.0D0
04510           DO 50 L=1,MAX
04520             50  PRJCT=PRJCT+EV(L,I)*EV(L,J)
04530             DO 60 L=1,MAX
04540               60  EV(L,I)=EV(L,I)-PRJCT*EV(L,J)
04550             40  CONTINUE
04560             VNORM=0.0D0
04570             DO 70 L=1,MAX
04580               70  VNORM=VNORM+EV(L,I)**2
04590             IF(VNORM.GT.1.0D-20)THEN
04600               VNORM=1.0D0/SQRT(VNORM)
04610               DO 80 L=1,MAX
04620                 80  EV(L,I)=EV(L,I)*VNORM
04630             ELSE
04640               DO 90 L=1,MAX
04650                 90  EV(L,I)=0.0D0
04660               IDGN=IDGN-1
04670               NORM(I)=0
04680             END IF
04690             30  CONTINUE
04700
04710 C*** CHECK ORTHOGONALITY
04720       DO 100 I=2,NUMVEC
04730         DO 100 J=1,I-1
04740           PRD=0.0D0
04750           DO 110 L=1,MAX
04760             110 PRD=PRD+EV(L,I)*EV(L,J)
04770           IF(ABS(PRD).LT.1.0D-10)GO TO 100
04780           WRITE(6,200)I,J
04790           200  FORMAT(' NON-ORTHOGONAL VECTORS AT',2I4)
04800           RETURN

```

```

04810 100 CONTINUE
04820 RETURN
04830 END
04840
04850 C***** CHECK OF THE EIGENVECTOR AND EIGENVALUE *****
04860
04870 SUBROUTINE CHECK(X,V,LIST1,LIST2,MAX,N,IPAIR,IBOND,
04880 & ZRTIO,BONDWT,PRD)
04890
04900 C*** VARIABLES MARKED @ SHOULD BE GIVEN FROM MAIN PROGRAM
04910 C X @ EIGENVECTOR TO BE CHECKED
04920 C V # H*X
04930 C LIST1,2 @ CONFIGURATIONS IN THE SPACE OF THE SPECIFIED SZ
04940 C MAX @ MATRIX DIMENSION
04950 C N @ NUMBER OF SITES
04960 C IPAIR @ PAIRS OF SITES CONNECTED BY BONDS
04970 C IBOND @ NUMBER OF BONDS
04980 C ZRTIO @ RATIO OF Jz TO Jxy
04990 C BONDWT @ EXCHANGE INTERACTION OF EACH BOND Jxy(ij)
05000 C PRD # X*H*X
05010
05020 IMPLICIT REAL*8(A-H,O-Z)
05030 INTEGER SITE1,SITE2
05040 DIMENSION X(MAX),V(MAX)
05050 DIMENSION LIST1(MAX),LIST2(2**N)
05060 DIMENSION IPAIR(IBOND*2),BONDWT(IBOND),ZRTIO(IBOND)
05070
05080 DO 10 I=1,MAX
05090 10 V(I)=0.0D0
05100 DO 20 K=1,IBOND
05110 SITE1=IPAIR(K*2-1)-1
05120 SITE2=IPAIR(K*2)-1
05130 IS1=2**SITE1
05140 IS2=2**SITE2
05150 IS=IS1+IS2
05160 WGHT=BONDWT(K)
05170 DIAG=WGHT*0.5D0*ZRTIO(K)
05180 DO 20 I=1,MAX
05190 IBIT=IAND(LIST1(I),IS1)+IAND(LIST1(I),IS2)
05200 IF(IBIT.EQ.0.OR.IBIT.EQ.IS)THEN
05210 V(I)=V(I)-DIAG*X(I)
05220 ELSE
05230 V(I)=V(I)+DIAG*X(I)
05240 V(I)=V(I)-
05250 & X(LIST2(IEOR(IEOR(LIST1(I),IS1),IS2))) *WGHT
05260 END IF
05270 20 CONTINUE
05280
05290 PRD=0.0D0
05300 DO 30 I=1,MAX
05310 30 PRD=PRD+V(I)*X(I)
05320 C WRITE(6,100)PRD
05330 100 FORMAT(' X*H*X IS',1PD16.8)
05340 C WRITE(6,110)
05350 110 FORMAT(' H*X(J)/X(J) RATIOS')
05360 C WRITE(6,111)(V(I)/X(I),I=2,MAX,MAX/19)
05370 111 FORMAT(6X,1PD15.8,1X,1PD15.8,1X,1PD15.8,1X,1PD15.8,1X,1PD15.8)
05380 RETURN
05390 END

```

Appendix. E 1次元での使用例

```

00010 C*****
00020 C***
00030 C***          1D HEISENBERG ANTIFERROMAGNET          ***
00040 C***                      AS A                      ***
00050 C***          TEST OF THE SUBROUTINES:              ***
00060 C***          SZ, LANCZS, INVITR, CG, ORTHG, CHECK    ***
00070 C***
00080 C***          1985/10/16                               ***
00110 C***
00120 C*****
00130
00140          PARAMETER (N=18,MAX=48620,IBOND=N)
00150          PARAMETER (ISTEP=5)
00160          IMPLICIT REAL*8 (A-H,O-Z)
00170          DIMENSION E(4),V(MAX,3)
00180          DIMENSION LIST1(MAX),LIST2(2*N)
00185          DIMENSION BONDWT(IBOND),IPAIR(2*IBOND),ZRTIO(IBOND)
00190          DIMENSION VO(MAX),V1(MAX),V2(MAX),DELM(MAX)
00200          DIMENSION B(MAX),X(MAX),NORM(3)
00220          DATA BONDWT/IBOND*(-1.0D0)/
00225          DATA ZRTIO/IBOND*1.0D0/
00230
00240          WRITE(6,100)
00250 100  FORMAT(' ***** 1D HEISENBERG ANTIFERROMAGNET ',
00260          &,' *****')
00270
00280 C*** SITE INFORMATION
00290          IPAIR(1)=1
00300          IPAIR(2*N)=1
00310          DO 10 I=2,2*N-1
00320 10  IPAIR(I)=I/2+1
00330          WRITE(6,110)IPAIR
00340 110  FORMAT(/20X,'(1) SITE INFORMATION  << IPAIR >>'/10(2X,2I3))
00350
00360 C*** CONFIGURATIONS IN THE SPACE OF Sz=0
00370          CALL SZ(LIST1,LIST2,MAX,N,0.0D0)
00380
00390 C*** EIGENVALUES
00400          WRITE(6,120)
00410 120  FORMAT(/20X,'(2) FOUR LOWEST EIGENVALUES'/
00420          &' INITIAL VECTOR          E(1)          E(2)          E(3)',
00430          &'          E(4)          ITERATION')
00440          EGRND=100.0D10
00450          DO 20 IV=17,MAX,MAX/4
00460          CALL LANCZS(N,IPAIR,IBOND,V2,V1,VO,DELM,ZRTIO,ISTEP,E,ITRAT,
00470          &          IV,LIST1,LIST2,MAX,BONDWT)
00480          WRITE(6,130)IV,E,ITRAT
00490 130  FORMAT(2X,I6,5X,4F14.8,2X,I6)
00500          IF(E(1).LT.EGRND)THEN
00510          EGRND=E(1)
00520          END IF
00530 20  CONTINUE
00540
00550 C*** EIGENVECTOR OF THE GROUND STATE
00560          WRITE(6,140)
00570 140  FORMAT(/20X,'(3) EIGENVECTOR OF THE GROUND STATE')
00580          DO 30 L=1,3
00590          IV=13+MAX/3*(L-1)
00600          WRITE(6,150)L,L
00610 150  FORMAT('/' ((3.,I1,')) INITIAL VECTOR NUMBER',I5,' -----'/)

```



```
00620      WRITE(6,160)
00630 160    FORMAT('      INFORMATION FROM INVITR AND CG')
00640      CALL INVITR(EGRND,IPAIR,IBOND,LIST1,LIST2,MAX,N,B,X,
00650      &      VO,V1,V2,ISTEP,IV,ZRTIO,BONDWT)
00660      WRITE(6,170)
00670 170    FORMAT(/'      INFORMATION FROM CHECK')
00680      CALL CHECK(X,VO,LIST1,LIST2,MAX,N,IPAIR,IBOND,ZRTIO,BONDWT)
00690      DO 40 J=1,MAX
00700 40      V(J,L)=X(J)
00710 30      CONTINUE
00720
00730 C*** DEGENERACY OF THE GROUND STATE
00740      CALL ORTHG(V,NORM,IDGN,MAX,3)
00750      WRITE(6,180)IDGN,NORM
00760 180    FORMAT(/' ((3.4)) DEGENERACY OF THE GROUND STATE ',I8/
00770      &'      NORM OF THE ORTHOGONALIZED VECTORS : ',3I3)
00780      END
```

```

1 //B5085B JOB ****,CLASS=B,NOTIFY=B5085
2 ***MAIN SYSTEM=S810,DEST=T
3 //
13 //SYSTIN DD *
JGP3001 B5085 ,LAST ACCESSED DATE=85.10.21 TIME=17.21.50
JDU4501 B5085B ,CMD , ,STEP WAS EXECUTED,RTN CODE=0000
JDU7951 STEP(CMD ) ) START 85.10.21(294) 17.26
JDU7961 STEP(CMD ) ) STOP 85.10.21(294) 17.28 CPU TIME=( 1MIN,13.79SEC) REAL STORAGE= 1536KB SYS STORAGE= 244KB
JDU7971 JOB (B5085B ) START 85.10.21(294) 17.26 CPU TIME=( 1MIN,13.79SEC)
JDU7981 JOB (B5085B ) STOP 85.10.21(294) 17.28 CPU TIME=( 1MIN,13.79SEC)

JSN=L22780 ST=0:01:13.79 VT=0:01:00.91 ET=0:02:18 SYSTEM=S810(V) I/D=(DISK:78,VIO:0)
MEMORY=1536KB EX-MEMORY=838KB EX-STORAGE=4KB DATABASE=(C:Y0,U:Y0)
NETWOR=00 FILE=(S:209KB,Y1,L:342KB,Y1) ACCT=(A,Y****,Y*****,Y*****)

>>USE ,@TEST.LOAD
>>MATRIX
>>MSL2
>>GO MAIN,ENTRY(MAIN),LIB(LIB.LOAD)
END
JET110611 USER COMMAND PROFILE BEING STORED

***** 1D HEISENBERG ANTIFERROMAGNET *****
(1) SITE INFORMATION << IPAIR >>
1 2 3 4 5 6 7 8 9 10 11
11 12 13 14 15 16 17 18 18 1

(2) FOUR LOWEST EIGENVALUES
INITIAL VECTOR E(1) E(2) E(3) E(4) ITERATION
17 -16.04549817 -15.56299927 -15.26096250 -14.94297247 60
12172 -16.04549817 -15.56299927 -15.26096411 -14.94297293 55
24327 -16.04549817 -15.56299927 -14.94297298 -14.20528714 55
36482 -16.04549817 -15.56299927 -15.26096405 -14.94297289 60

(3) EIGENVECTOR OF THE GROUND STATE

(3.1) INITIAL VECTOR NUMBER 1 -----
INFORMATION FROM INVTR AND CG
NUMBER OF ITERATIONS IN CG : 80
NUMBER OF ITERATIONS IN CG : 115
NUMBER OF ITERATIONS IN CG : 30
NUMBER OF ITERATIONS IN INVTR : 3
INFORMATION FROM CHECK
X=H*X IS -1.60454982D+01
H*(X)/X(R) RATIOS
-1.60454982D+01 -1.60454982D+01 -1.60454982D+01 -1.60454982D+01
-1.60454982D+01 -1.60454982D+01 -1.60454982D+01 -1.60454982D+01
-1.60454982D+01 -1.60454982D+01 -1.60454982D+01 -1.60454982D+01
-1.60454982D+01 -1.60454982D+01 -1.60454982D+01 -1.60454982D+01
-1.60454982D+01 -1.60454982D+01 -1.60454982D+01 -1.60454982D+01

(3.2) INITIAL VECTOR NUMBER 2 -----
INFORMATION FROM INVTR AND CG

```

```
NUMBER OF ITERATIONS IN CG      :   95
NUMBER OF ITERATIONS IN CG      :   60
NUMBER OF ITERATIONS IN INVITR  :    2
```

INFORMATION FROM CHECK

X*H*X IS -1.60454982D+01

H*X(J)/X(J) RATIOS

```
-1.60454982D+01 -1.60454982D+01 -1.60454982D+01 -1.60454982D+01
-1.60454982D+01 -1.60454982D+01 -1.60454982D+01 -1.60454982D+01
-1.60454982D+01 -1.60454982D+01 -1.60454982D+01 -1.60454982D+01
-1.60454982D+01 -1.60454982D+01 -1.60454982D+01 -1.60454982D+01
-1.60454982D+01 -1.60454982D+01 -1.60454982D+01 -1.60454982D+01
```

((3.3)) INITIAL VECTOR NUMBER 3 -----

INFORMATION FROM INVITR AND CG

```
NUMBER OF ITERATIONS IN CG      :   95
NUMBER OF ITERATIONS IN CG      :   60
NUMBER OF ITERATIONS IN INVITR  :    2
```

INFORMATION FROM CHECK

X*H*X IS -1.60454982D+01

H*X(J)/X(J) RATIOS

```
-1.60454982D+01 -1.60454982D+01 -1.60454982D+01 -1.60454982D+01
-1.60454982D+01 -1.60454982D+01 -1.60454982D+01 -1.60454982D+01
-1.60454982D+01 -1.60454982D+01 -1.60454982D+01 -1.60454982D+01
-1.60454982D+01 -1.60454982D+01 -1.60454982D+01 -1.60454982D+01
-1.60454982D+01 -1.60454982D+01 -1.60454982D+01 -1.60454982D+01
```

((3.4)) DEGENERACY OF THE GROUND STATE 1
NORM OF THE ORTHOGONALIZED VECTORS : 1 0 0

Appendix. F

行列次元 1500 以下の場合の MRXELM の使用例

```

00010      IMPLICIT REAL*8 (A-H,O-Z)
00020      PARAMETER (N=4,MAX=6 ,IBOND=3)
00030 C
00040      DIMENSION IPAIR(IBOND*2),ELEMNT(MAX*(MAX+1)/2),BONDWT(IBOND)
00050      DIMENSION LIST1(MAX),LIST2(2**N),ZRTIO(IBOND)
00060      DIMENSION WK(MAX,6),IWK(MAX),E(MAX)
00070 C
00080      DATA BONDWT/IBOND*1.0D0/
00090      DATA ZRTIO /IBOND*1.0D0/
00100      DATA IPAIR /1,2, 2,3, 3,4/
00110      CALL SZ(LIST1,LIST2,MAX,N,0.0D0)
00120 C
00130      CALL MRXELM(MAX,IPAIR,IBOND,LIST1,LIST2,N,BONDWT,ELEMNT,ZRTIO)
00140 C
00142      MX=MAX*(MAX+1)/2
00150      CALL %DEF4M(ELEMNT,MAX,MX,MAX,1,-0.1,3,E,V,MAX,IFLG,WK,IWK,IER)
00160      WRITE(*,*) (E(I),I=1,MAX)
00170      STOP
00180      END
00190 C***** CONFIGURATIONS WITH SPECIFIED SZ *****
00200
00210      SUBROUTINE SZ(LIST1,LIST2,MAX,N,SZVAL)
00220
00230 C*** VARIABLES MARKED @ SHOULD BE GIVEN FROM MAIN PROGRAM
00240 C*** VARIABLES MARKED # ARE EVALUATED AND RETURNED
00250 C      LIST1(I)  # I-TH SPIN CONFIGURATION EXPRESSED BY BIT PATTERN
00260 C      LIST2      # INVERSE LIST OF LIST1
00270 C      MAX        @ DIMENSION OF THE MATRIX
00280 C      N          @ NUMBER OF SITES OF THE LATTICE
00290 C      SZVAL      @ TOTAL SZ OF THE SPACE TO BE DIAGONALIZED
00300
00310      REAL*8 SZVAL
00320      DIMENSION LIST1(MAX),LIST2(2**N)
00330
00340      IF(MAX.GT.184756.OR.N.GT.20)THEN
00350        WRITE(6,*)' TOO LARGE MAX OR N GIVEN TO SZ'
00360        STOP
00370      END IF
00380      IF(SZVAL.LT.-1.0D-20)THEN
00390        WRITE(6,*)' SZVAL SHOULD NOT BE NEGATIVE'
00400        STOP
00410      END IF
00420      IF(SZVAL.GT.N/2.D0+1.D-20)THEN
00430        WRITE(6,*)' SZVAL SHOULD NOT EXCEED N/2'
00440        STOP
00450      END IF
00460      IUPSPN=N/2+MOD(N,2)+INT(SZVAL+0.001D0)
00470      ICNT=0
00480      DO 10 I=1,2**N
00490        ISZ=0
00500        LIST2(I)=0
00510        DO 20 J=0,N-1
00520 20      ISZ=ISZ+MOD(I/2**J,2)
00530        IF(ISZ.NE.IUPSPN)GO TO 10
00540        ICNT=ICNT+1
00550        IF(ICNT.GT.MAX)THEN
00560          WRITE(6,*)' INCORRECT MAX OR N GIVEN TO SZ'
00570          STOP
00580        END IF
00590        LIST1(ICNT)=I

```

```

00600          LIST2(I)=ICNT
00610 10      CONTINUE
00620          IF(ICNT.EQ.MAX)RETURN
00630          WRITE(6,*)' INCORRECT MAX OR N GIVEN TO SZ'
00640          STOP
00650          END
00660
00670          SUBROUTINE MRXELM(MAX,IPAIR,IBOND,LIST1,LIST2,N,BONDWT,ELEMNT,
00680 &              ZRTIO)
00690
00700 C*** VARIABLES MARKED @ SHOULD BE GIVEN FROM MAIN PROGRAM
00710 C*** VARIABLES MARKED # ARE EVALUATED AND RETURNED
00720 C      MAX          @ MATRIX DIMENSION
00730 C      IPAIR        @ PAIRS OF SITES CONNECTED BY BONDS
00740 C      IBOND        @ NUMBER OF BONDS
00750 C      LIST1,2      @ CONFIGURATIONS IN THE SPACE OF THE SPECIFIED SZ
00760 C      N            @ NUMBER OF SITES
00770 C      BONDWT       @ EXCHANGE INTERACTION OF EACH BOND Jxy(ij)
00780 C      ELEMNT      # MATRIX ELEMENTS TO BE RETURNED
00790 C      ZRTIO       @ RATIO OF Jz TO Jxy
00800
00810          IMPLICIT REAL*8 (A-H,O-Z)
00820          INTEGER SITE1,SITE2
00830          DIMENSION IPAIR(IBOND*2),ELEMNT(MAX*(MAX+1)/2),BONDWT(IBOND)
00840          DIMENSION LIST1(MAX),LIST2(2**N),ZRTIO(IBOND)
00850
00860 C*** INITIALIZATION
00870          DO 10 I=1,MAX*(MAX+1)/2
00880 10      ELEMNT(I)=0.0D0
00890
00900 C*** ELEMENTS
00910          DO 30 K=1,IBOND
00920              SITE1=IPAIR(K*2-1)-1
00930              SITE2=IPAIR(K*2 )-1
00940              IF(SITE1.LE.-1.OR.SITE1.LE.-1)THEN
00950                  WRITE(6,*)' INCORRECT DATA IN IPAIR FOUND IN MRXELM'
00960                  STOP
00970              END IF
00980              IS1=2**SITE1
00990              IS2=2**SITE2
01000              IS=IS1+IS2
01010              WGHT=BONDWT(K)
01020              DIAG=WGHT*0.5D0*ZRTIO(K)
01030 *VOPTION VEC
01040              DO 30 I=1,MAX
01050                  IBIT=IAND(LIST1(I),IS1)+IAND(LIST1(I),IS2)
01060                  IDG=I*(I+1)/2
01070                  IF(IBIT.EQ.0.OR.IBIT.EQ.IS)THEN
01080                      ELEMNT(IDG)=ELEMNT(IDG)-DIAG
01090                  ELSE
01100                      ELEMNT(IDG)=ELEMNT(IDG)+DIAG
01110                      IEXCHG=LIST2(IEOR(IEOR(LIST1(I),IS1),IS2))
01120                      IF(IEXCHG.GE.I)GO TO 30
01130                      ELEMNT(I*(I-1)/2+IEXCHG)=-WGHT
01140                  END IF
01150 30      CONTINUE
01160          RETURN
01170          END

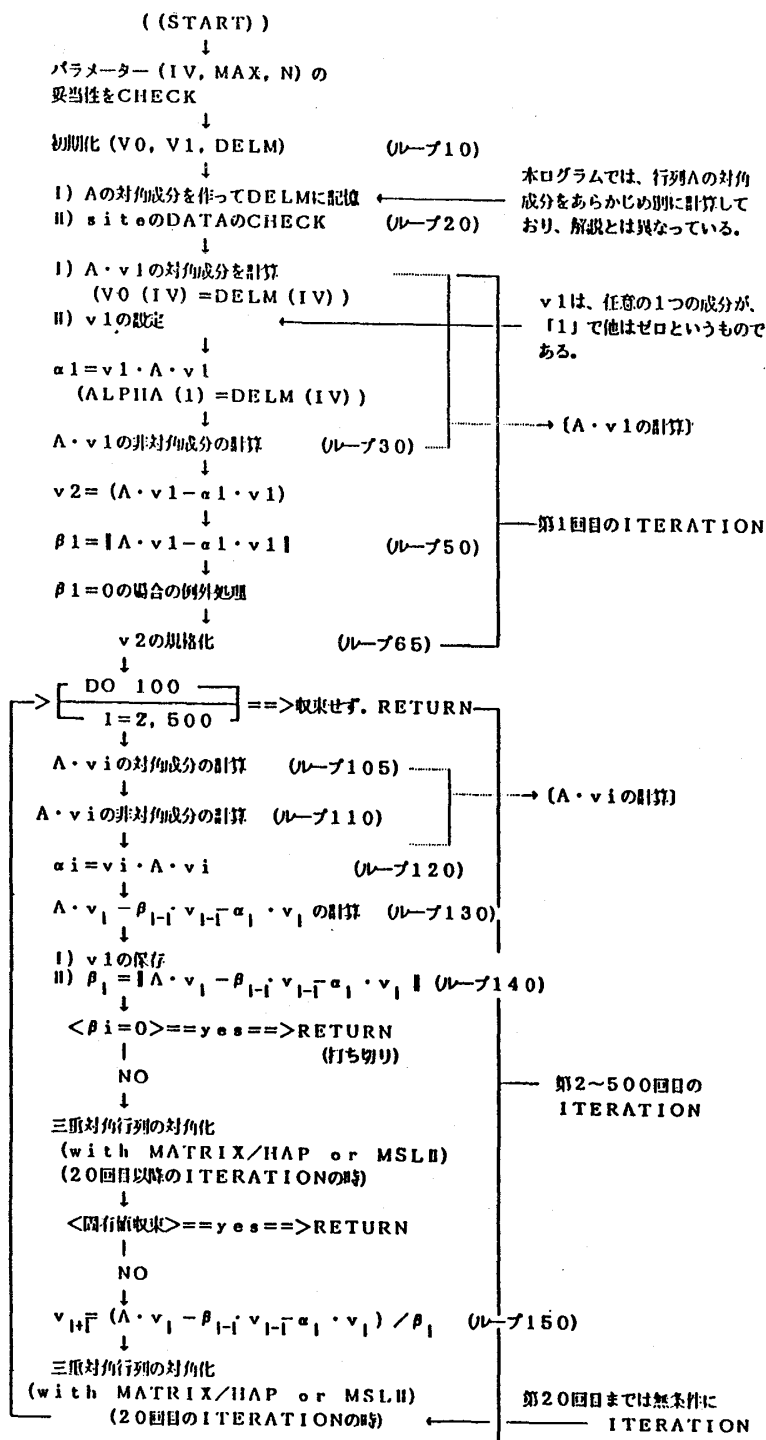
```

Appendix. G

主要サブルーチンのフロー・チャート

【フロー・チャート LANCZS】 (固有値を求めるサブルーチン)

[ref. 2のp169 ①~⑥に従っている。ハミルトニアン行列 H を A と表記している。]



【フロー・チャート, CG】

(CG法)

[ref. 2のp95~p97
アルゴリズム①~③, ④, ⑥~⑦
に従っている。]

((START))

ref. 2のアルゴリズム・ナンバー

I) 初期設定, $R=B$, $P=B$

II) 初期化 $X=0$

→① ($X=0$ とおいたことに相当)

(ループ10)

DO 20
ITERAT=1, MAX

初期化 $Y_k = 0$ (ループ30) →①

$Y_k = A \cdot P_k$ (ループ40)

I) $RP = R_k \cdot P_k$

II) $YP = Y_k \cdot P_k (= P_k \cdot A \cdot P_k)$ (ループ50)

$\alpha_k = \frac{(P_k, R_k)}{(P_k, A \cdot P_k)} = \frac{RP}{YP}$

I) $X_{k+1} = X_k + \alpha_k \cdot P_k$ →③

II) $RNORM = R_k \cdot R_k$ (ループ60)

I) $R_{k+1} = R_k - \alpha_k \cdot Y_k$ →④

II) $RNORM2 = R_{k+1} \cdot R_{k+1}$ (ループ70) →⑤

$\beta_k = \frac{(R_{k+1}, R_{k+1})}{(R_k, R_k)} = \frac{RNORM}{RNORM2}$ (2.65) 式

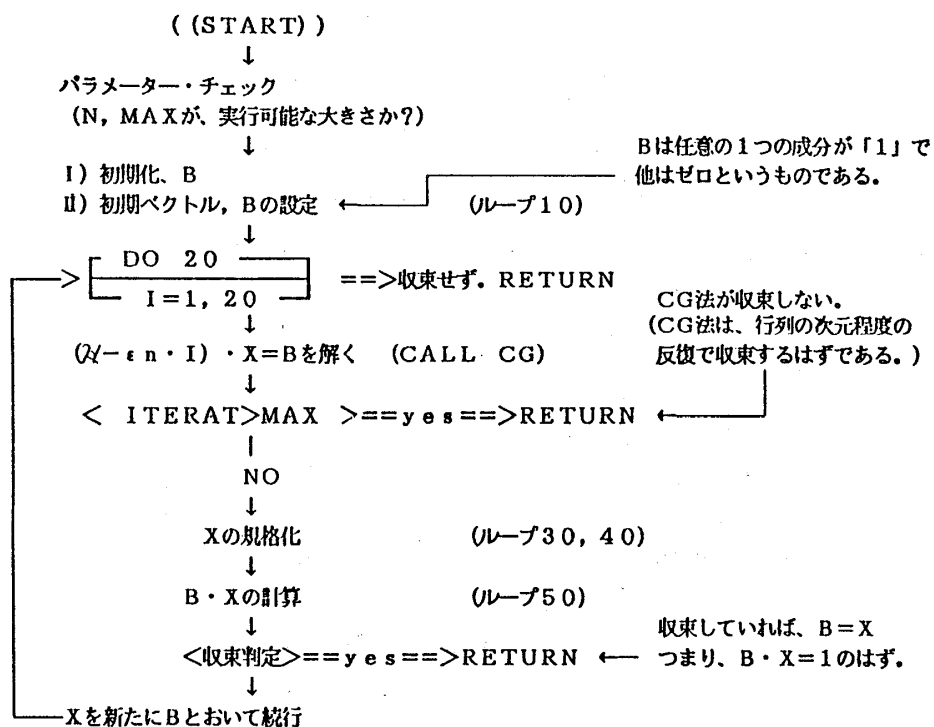
$P_{k+1} = R_{k+1} + \beta_k \cdot P_k$ (ループ90) →⑥

NO → <収束判定> (2.66) 式 (但し, $\epsilon = 0.5 \times 10^{-6} \times ||B||$)

YES

RETURN

《フロー・チャート INVITR》 (逆反復法のサブルーチン)



Appendix. H

ビット演算解説

本文 § 4 に書いた通り、本プログラムの本質は $\lambda \cdot \phi$ を、行列 λ を生成しながら、計算するという部分にある。この部分の説明はかなりややこしいので、フロー・チャートでも省いている。よって、ここで具体的に説明しよう。(ここを読む前に必ず § 4 を熟読すること) 例を「LANCZS」の DO ループ、20 (対角成分)、DO ループ、30 (非対角成分) にとって、説明することにする。

1) DO ループ、20 (対角成分)

SITE1=IPAIR(I*2-1)-1

SITE2=IPAIR(I*2)-1

site1, site2 というのは、相互作用している SITE つまり、ハミルトニアン (1) の $\langle i, j \rangle$ である。1 を引いてあるのは、後で使うビット演算関数の表式で、2 進数の一番下の桁は 0 bit 目と名付けられているからである。つまり、site1 は、0 bit 目、site2 は 1 bit 目、というふうに対応させたいのである。そうしないと § 4 で述べた、2 進数と直交関数系の対応がおかしくなってしまう。

IS1=2**SITE1

IS2=2**SITE2

これにより、IS1, IS2 は対応する bit のところのみ 1 で後の bit は全て 0 という数になる。

(中略)

IS=IS1+IS2

IS は、2 つの相互作用している bit に各々 1 が入っていて、他は 0 という数になる。

(SITE1 \neq SITE2 に注意)

(中略)

IBIT=IAND(LIST1(J), IS1)+IAND(LIST1(J), IS2)

詳しいことは省くが、IAND (a1, 2**n) とやると、a1 の n-bit 目 (n=0, 1, 2, ...) の値 (0 か 1) だけがそのまま残り、後の bit は全てゼロ、という数が与えられる。それ故 IBIT のとりうる値は、LIST1 (J) の値によって、0, IS1, IS2, IS1+IS2 (=IS) , の 4 通りだけである。

IF (IBIT.EQ.0.OR.IBIT.EQ.IS) THEN

```

PARREL=0.5D0
ELSE
PARREL=-0.5D0
END IF

```

今、計算しているのは、対角成分であるから、ハミルトニアン (1) の $S_i^z S_j^z$ の部分である。この値は、i-site と j-site の spin が平行か反平行かで決まる。IBIT の値をみれば、どちらであるか解る。IBIT=0 or IS なら、平行なので、 $1/2$ 、IBIT=IS1 or IS2 なら、反平行なので、 $-1/2$ を入れる。 $1/4$ 、 $-1/4$ 、でないのは、ハミルトニアン (1) のファクター「2」をいっしょに計算してあるからである。

```
DELM(J)=DELM(J)-WGHT*PARREL
```

WGHT には $J_{ij} \cdot \Delta_{ij}$ の値が入っているので、これを計算すれば、 $S_i^z S_j^z$ という項の第 J 対角要素への寄与が計算できる。

II) DO ループ、30 (非対角成分)

```

SITE1=IPAIR(K*2-1)-1
SITE2=IPAIR(K*2 )-1
IS1=2**SITE1
IS2=2**SITE2
IS=IS1+IS2
IBIT=IAND(LIST1(I), IS1)+IAND(LIST1(I), IS2)

```

ここまでは、I) DO ループ、20 と同じである。

```
IF (IBIT.EQ.0.OR.IBIT.EQ.IS) GOTO 30
```

今度は、非対角成分であるから $1/2 \cdot (S_i^+ S_j^- + S_i^- S_j^+)$ の部分である。 $S=1/2$ なので、すぐ解るように、i-site の spin と j-site の spin は反平行でなくてはならない。I) で述べたように、IBIT=0, or IS の時は、spin が平行なので、この時は処理をしないでとばす。それでは反平行の時はどうするかというと

```
IEXCHG=IEOR(IEOR(LIST1(I), IS1), IS2)
```

またまた、細かいことは省くが、IEOR (a1, 2**n) とやると、a1 という数の n-bit 目が逆転した ($1 \rightarrow 0$, $0 \rightarrow 1$) 数が与えられる。つまり、IEXCHG には i-site と j-site の spin が逆転した状態を示す 2 進数が入っていることになる。この 2 進数が、 $S_i^+ S_j^- + S_i^- S_j^+$ を作用させた後の状態を表わすことになるので、

```
V0(I)=V0(I)-V1(LIST2(IEXCHG))*WGHT
```

をやれば、ベクトル V1 と行列の非対角成分との積が求まるわけである。ファクター $1/2$ は、I) の時と同じく、ファクター 2 と相殺している。CG でやっていることも、本質的には同じ

ことなので、ここでは説明しない。しかし、ひとつだけ解りにくいところがあるので、注意しておく。それは、CGのDOループ、40の中の「 $EPERBD = E / \text{FLOAT}(\text{IBOND})$ 」のところである。なぜ、IBONDで割るのか？。もし割らないとすると、 $\sum (\text{IBOND}) \times \epsilon_n \cdot I$ になってしまうと言え、解って頂けることと思う。最後に、ひとつだけ述べておく。お気付きの方もいるかもしれないが、このbit演算はBTESTやIBCHNGという組み込み関数を用いれば、もっと能率よく行なうことができるのである。しかし、HITAC FORT77/HAPのコンパイラは今のところ、この関数をベクトル化してくれない。(引数が定数だとベクトル化する。) そのうち、コンパイラが修正されることもあるかもしれないが、我々がプログラムを開発した時点(1985年 2~5月)ではまだ駄目であった。プログラムをいじって速くしたいと思われる方は、その辺にご留意頂きたい。